4

Technical Note 8-89

SOFTWARE AND HARDWARE DESCRIPTION OF THE HELICOPTER MOTION EQUATIONS FOR VAX COMPUTERS

Maria del C. Lopez

August 1989
AMCMS Code 612716H7040

Approved for public release;
distribution is unlimited.

U. S. ARMY HUMAN ENGINEERING LABORATORY

Aberdeen Proving Ground, Maryland

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): Technical Note 8-89	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Human Engineering Laboratory — 6b OFFICE SYMBOL *(If applicable)*: SLCHE	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)*: Aberdeen Proving Ground, MD 21005-5001	7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
6.2	1L162716AH70		

11 TITLE *(Include Security Classification)*
Software and Hardware Description of the Helicopter Motion Equations for VAX Computers

12. PERSONAL AUTHOR(S)
Lopez, Maria del C.

13a. TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Final	FROM ______ TO ______	1989, August	92

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
01	03	01
(see reverse side)		

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
human engineering, flight equations, helicopter, cockpit, UH 60, displays, simulator, Blackhawk (see reverse side)

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report describes the software design and hardware configuration used to execute a set of helicopter motion equations on a MicroVAX II VAXLab computer under the VMS operating system in a real-time, manned, interactive simulation. The equations were acquired from NASA Ames, California, with the objective of simulating aerodynamic characteristics of a helicopter in a part-task simulator for human factors studies related to displays and controls. The software design includes four modules that are run simultaneously to execute the equations. The four modules are explained in detail and a listing of source codes are included in the appendixes.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL: Technical Reports Office	22b TELEPHONE *(Include Area Code)*: (301) 278-4478 — 22c. OFFICE SYMBOL: SLCHE-SS-IR

17. (continued)

12 05
12 06

18. (continued)

controls
hardware
software
VAX
FORTRAN
real time

AMCMS Code 612716H7040 Technical Note 8-89

SOFTWARE AND HARDWARE DESCRIPTION OF THE HELICOPTER MOTION EQUATIONS FOR VAX COMPUTERS

Maria del C. Lopez

August 1989

APPROVED: John D. Weisz
JOHN D. WEISZ
Director
Human Engineering Laboratory

Approved for public release;
distribution is unlimited.

U.S. ARMY HUMAN ENGINEERING LABORATORY
Aberdeen Proving Ground, Maryland 21005-5001

CONTENTS

INTRODUCTION 3

SYSTEM DESCRIPTION 3

GLOBAL SECTION DESCRIPTION 4

PROCESS DESCRIPTIONS

START_HACSEC Process 7
AD30HZ Process 8
DR11_SND_RCV Process 8
HACMAIN_BH Process 9

SEQUENCE DESCRIPTION 10

CONCLUSIONS 11

REFERENCES 13

APPENDIXES

A. Include Files Source Code 15
B. Main Programs Source Code 31
C. Subroutine Source Code 47

FIGURES

1. Hardware Configuration 5
2. Software Configuration 6

SOFTWARE AND HARDWARE DESCRIPTION OF THE HELICOPTER MOTION EQUATIONS

FOR VAX COMPUTERS

INTRODUCTION

The U.S. Army Human Engineering Laboratory (HEL) at Aberdeen Proving Ground, Maryland, is the U.S. Army Laboratory Command's lead laboratory for human factors engineering. Increasingly sophisticated equipment and soldier interfaces require the laboratory to perform effective investigations of complex soldier-machine interfaces.

The Human Factors Cockpit Research, Experimentation, and Workload (CREW) Simulator is a fixed base, generic helicopter simulator that provides the flexibility to explore state-of-the-art aircrew/display/control interaction. The CREW simulator involves four major components: the visual system, the graphics system, the data collection, and the flight equations. HEL uses the UH60 Blackhawk version of the standard kinematic equations for an aircraft (herein called HAC equations) when conducting experiments on the CREW simulator.

The HAC equations were acquired from NASA Ames, in Mountain View, California, with the objective of simulating aerodynamic characteristics of a helicopter in a part-task simulator for human factors studies related to displays and controls. The HAC equations also allow us to simulate nap-of-the-earth (NOE) flight. The HAC equations are based on the second-order Adams-Bashford predictor integration method and the modified Euler algorithm. They do not model the engine or rotor systems of the helicopter.

This report describes the software design and the hardware configuration used to execute the HAC equations on a MicroVAX II VAXLab computer under the VMS operating system using VAX FORTRAN. The HAC equations are not described in detail in this report. For more detailed information, refer to McFarland (1975). The objective of this report is to provide internal documentation and also to provide a description of design for others desiring to implement these or similar equations of motion under the VAX/VMS (Virtual Address eXtension/Virtual Memory System) operating system in a real-time[1], manned, interactive simulation. (KR)

SYSTEM DESCRIPTION

Hardware Configuration

The HAC equations are run on a Digital Equipment Corporation MicroVAX II VAXLab with 5.0 megabytes (MB) of memory. The following hardware devices, which are interfaced to the MicroVAX II VAXLab, are used for the HAC equations:

[1]Real-time in simulation is executing a process at a speed fast enough to give a response within the actual time of the real event.

- ADV11-DA digital-to-analog converter
- DRV11-W communications board
- conventional helicopter controls and four-axis controls

The output from the HAC equations is sent to the VAX 11/780 for cockpit out-the-window visuals and from the VAX 11/780 to the VAX 11/750 for graphics displays and performance data collection. The MicroVAX II VAXLab uses an Ethernet/DECnet communications network to start processes for visuals and displays on the VAX 11/780 and the VAX 11/750. Figure 1 shows a general picture of the hardware configuration of the system (Herald, 1987).

Software Configuration

The HAC equations are run under the MicroVAX VMS Version 4.4 operating system. The source programs for the HAC equations are written in the VAX FORTRAN language Version 5.0 (Digital Equipment Corporation, 1984) except for two modules written in VAX MACRO.

Four processes[2] are executed simultaneously when running the HAC equations. Therefore, priorities need to be set to determine the order in which these executable processes are to run. The four processes are START_HACSEC, which creates the global section; AD30HZ, which reads the analog values from the cockpit controls and digitizes their values; HACMAIN_BH, which executes the HAC equations; and DR11_SND_RCV, which transmits the outputs to the VAX/780. Figure 2 shows the system when the processes are running. In general 32% of the central processing unit (CPU)[3] time on the MicroVAX II VAXLab is used.

All processes use the VAX system services to create and map the global section, to control the execution speed of the process, and to communicate between processes through event flags. System services are procedures that the VAX/VMS operating system uses to

- control resources available to processes;

- provide for communication among processes; and

- perform basic operating system functions, such as the coordination of input/output operations.

GLOBAL SECTION DESCRIPTION

A global section is either a disk file or a page-file section containing shareable code or data. The global section referenced in this report is a disk file.

[2]A process is the execution of a program image.
[3]CPU time is the amount of the CPU consumed by the program when it is executed.

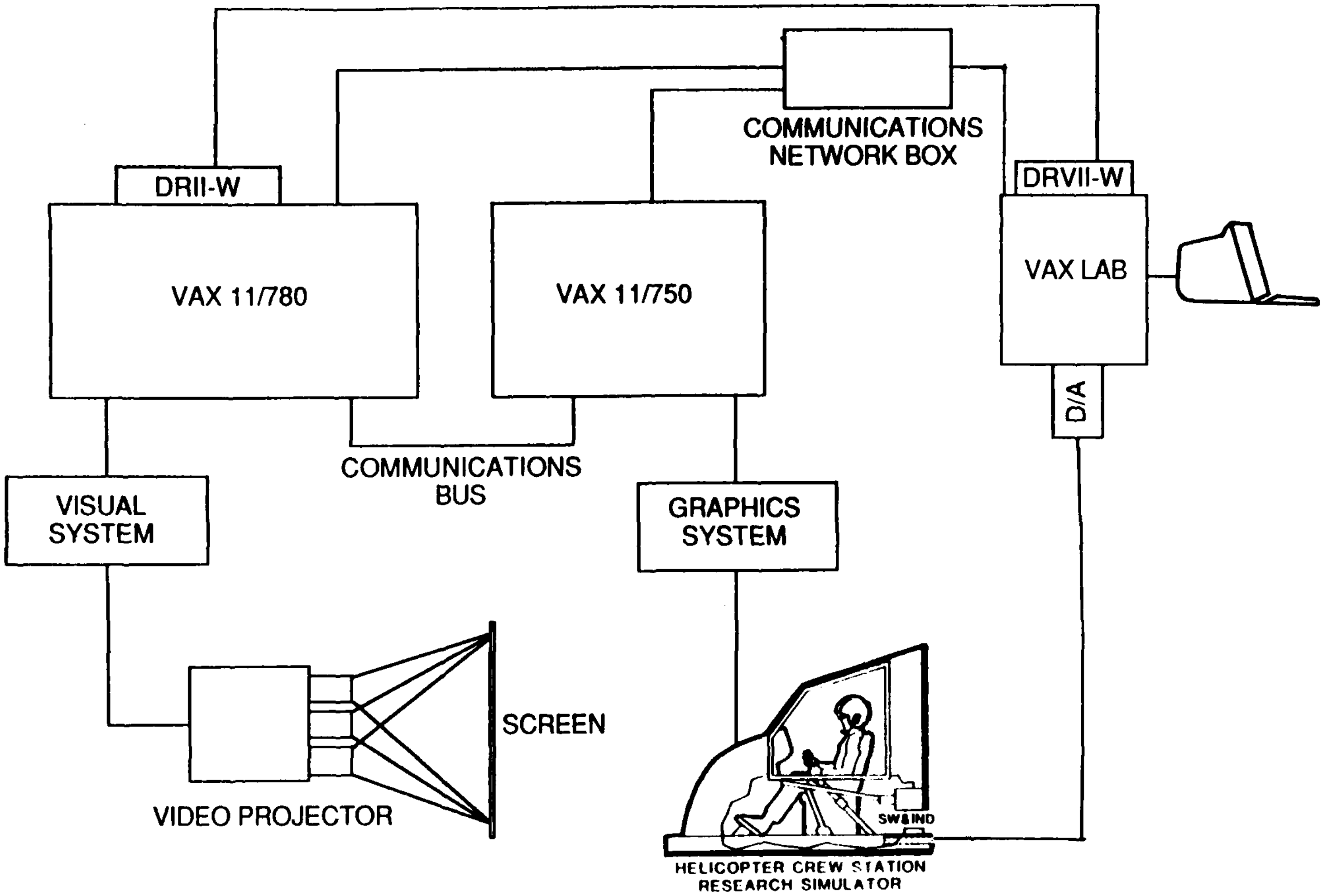

Figure 1. Hardware configuration.

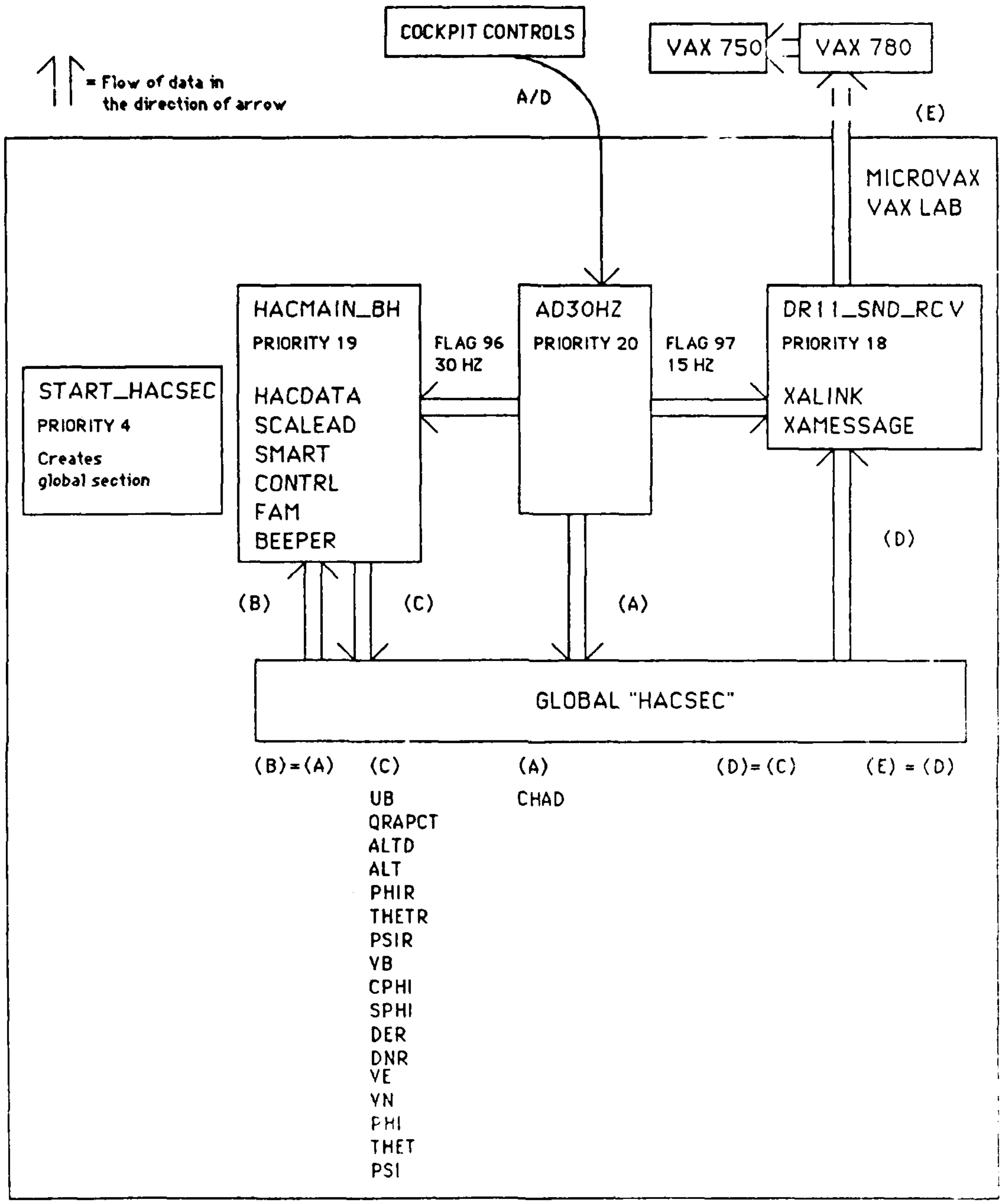

Figure 2. Software configuration.

The following characteristics apply to the global section:

a. The starting and ending virtual addresses are the address of the variable ARM(1) and the address of the variable WAYPT_ERR, respectively.

b. The global section is set up as a temporary group global section, with read and write access to the pages, and all pages initialized to zero when created.

c. The global section's name is HACSEC.

d. The name of the disk file opened to map the pages is HACSEC.DAT.

VAX FORTRAN common statements are used in all source programs for the HAC equations to arrange the variables in the global section HACSEC. A common statement defines one or more contiguous areas, or blocks of storage. Common statements also define the order in which variables and arrays are stored in each common block. Appendix A lists the source code that has all commons included in the four programs, which are START_HACSEC, AD30HZ, DR11_SND_RCV and HACMAIN_BH.

PROCESS DESCRIPTIONS

START_HACSEC Process

Description

The START_HACSEC process is responsible for creating and mapping to the global section HACSEC shared by HACMAIN_BH, AD30HZ, and DR11_SND_RCV processes.

The START_HACSEC process runs only once and suspends itself, thus keeping the global section in memory and accessible to the other processes for mapping. This technique is useful in making HACMAIN_BH, AD30HZ, and DR11_SND_RCV processes independent of each other allowing individual process priorities and scheduling. The process is kept at a non-real-time priority 4 and is the first one to run. The listing of the source code is in Appendix B.

Output

The output to this process is the created global section with all the variables that appear in the include files[4] listed in Appendix A.

[4]An include file is a module of source text that can be incorporated into a FORTRAN program by using the INCLUDE FORTRAN statement.

AD30HZ Process

Description

The AD30HZ process is responsible for controlling the analog-to-digital converter and converting analog values from the cockpit controls to digital values. The AD30HZ process uses the synchronous user interface LIO$READ from SYS$LIBRARY:LIOSET.FOR symbol definitions. The process runs at real-time priority 20. The process reads the analog values for 16 channels via the ADV11 interface 30 times every second and puts the values in a buffer from which only four channels are put in the global section HACSEC. The four channels used represent pitch, roll, collective and pedals digital outputs. Pitch and roll analog values come from the four-axis controller interface; the collective and pedals analog values come from the conventional helicopter collective and pedals controls. The remaining 12 channels are not presently used.

The AD30HZ process is also responsible for controlling the processing speed of HACMAIN_BH and DR11_SND_RCV through event flags 96 and 97 respectively on VAX cluster 3. This technique is used to optimize the CPU time used by the processes and at the same time it assures sending the latest values to the VAX 11/780 for processing. The listing of the source code appears in Appendix B.

Input

The four values read from the analog channels serve as the input to this process.

Output

The four digital values are put in the global section through the variable CHAD.

DR11_SND_RCV Process

Description

The DR11_SND_RCV process is responsible for sending the values computed by HACMAIN_BH equations to the VAX 11/780. The process runs at real-time priority 18. The values are obtained from the global section HACSEC and are sent via the DR11W interface 15 times every second. The values that are sent to the VAX 11/780 are used to update the visual system. The listing of the main program source code appears in Appendix B.

DR11_SND_RCV is divided into the following MACRO and FORTRAN subroutines[5]:

[5]A FORTRAN subroutine is a program unit consisting of a SUBROUTINE statement followed by a series of statements that define a computing procedure. A CALL statement is used to transfer control to a subroutine and a RETURN statement is used to return control to the calling program unit.

- UPDATE_TO_VAX--Updates variables that are going to be sent to the VAX 11/780

- UPDATE_FROM_VAX--Updates variables with values sent from the VAX 11/780

- DR11_SND_RCV--Sends and receives values from or to the VAX 11/780

- XALINK and XAMESSAGE--Interface with DR11 drivers[6] using DEC MACRO subroutines

The DR11_SND_RCV process loops through the above modules 15 times every second controlled by the AD30HZ process through event flag 97. The listings of the MACRO and FORTRAN subroutines are in Appendix C.

Input

The values that serve as input are airspeed (UB), torque (QRAPCT), altitude (ALT), rate of climb (ALTD), pitch in radians (THETR), pitch in degrees (THET), roll in radians (PHIR), roll in degrees (PHI), yaw in radians (PSIR), roll in degrees (PSI), cosine and sine of roll (CPHI and SPHI), velocity east (VE), and velocity north (VN).

Output

The same values that serve as input are also output.

HACMAIN_BH Process

Description

The HACMAIN_BH process is responsible for running the HAC equations. The equations are based on the second-order Adams-Bashford predictor integration method and the modified Euler algorithm.

All axis systems are orthogonal[7], right-handed triads. The axis systems used in the equations are the Earth frame (E-frame), the Local frame (L-frame), and the Body frame (B-frame).

The E-frame is an inertial frame with the origin of the coordinates at the Earth's center; the Z_e axis intersects the North Pole, and the X_e axis intersects the zero-degree longitude line (Greenwich) at "zero time." The L-frame is situated on the Earth's surface, directly under the vehicle. Its X_l axis points northward and its Y_l axis points eastward; both are parallel to the

[6]A device driver is a set of routines and tables that the system uses to process an input/output request for a particular device type.
[7]Intersecting or lying at right angles.

Earth's surface. The Z_1 axis points toward the Earth's center. The L-frame follows the motion of the aircraft but its distance from the center of the Earth is constant. The B-frame uses the conventional aircraft notation; the X_b axis passes through the nose of the vehicle, the Y_b axis points toward the right wing, and the Z_b axis passes through the bottom of the vehicle. The B-frame origin is located at the vehicle's center of gravity. The listing of the main program is in Appendix B.

HACMAIN_BH is divided into the following subroutines (The listings of these subroutines are in Appendix C.):

- HACDATA--Initializes the values

- SCALEAD--Scales the analog-to-digital values read by the AD30HZ process

- SMART--Solves the force components in Earth frame, altitude, accelerations, velocities, rotational rates, Euler angles, eyepoint position, turbulence, and airspeed

- CONTRL--Provides for helicopter controls

- FAM--Provides helicopter forces, accelerations, and moments including aero, engine, and part of the equations of motion. It calculates target rotational accelerations, total translational forces on the vehicle and limits calculations for Z force

- BEEPER--Provides a trim routine

The HACMAIN_BH process loops through the above modules 30 times every second controlled by the AD30HZ process through event flag 96. It runs at real-time priority 19.

Input

The same four channels (pitch, roll, collective, and pedals) that are output from the AD30HZ process (CHAD) are output for the HACMAIN_BH process.

Output

The values that serve as output are airspeed (UB), torque (QRAPCT), altitude (ALT), rate of climb (ALTD), pitch in radians (THETR), pitch in degrees (THET), roll in radians (PHIR), roll in degrees (PHI), yaw in radians (PSIR), roll in degrees (PSI), cosine and sine of roll (CPHI and SPHI), velocity east (VE), and velocity north ([illegible]).

SEQUENCE DESCRIPTION

The first process to be run is START_HACSEC since it creates the global section HACSEC for the other three processes. Once the global section is

created, the other three processes may start running in any order. It is a good idea to run the process with the lowest priority first to make sure it is created with no delay. Following this, the order would be DR11_SND_RCV with priority 18, HACMAIN_BH with priority 19, and AD30HZ with priority 20. DR11_SND_RCV and HACMAIN_BH will not start looping through the program until AD30HZ starts running and setting the respective event flags.

CONCLUSIONS

The HAC equations software arranged in four modules makes it easier for the user to make modifications. These modules may be used separately on other applications since they are written in a general form.

When running the HAC equations software on the VAXLab, only 32% of the CPU is used. This gives 68% of the CPU time to run other HEL applications for testing without affecting the performance of the equations.

Using more than one computer to perform investigations of complex soldier-machine interfaces is an approach taken by HEL to support the wide range of tasks for the simulation needs and to allow for easy growth at a low initial cost.

REFERENCES

Digital Equipment Corporation. (1984). Programming in VAX FORTRAN version 4.4. Maynard, MA: Digital Equipment Corporation.

Digital Equipment Corporation. (1986). VAX/VMS system services reference manual version 4.4. Maynard, MA: Digital Equipment Corporation.

Herald, G. L. (1987). Human factors research simulator (Technical Memorandum 8-87). Aberdeen Proving Ground, MD: U.S. Army Human Engineering Laboratory.

McFarland, R. E. (1975). A standard kinematic model for flight simulation at NASA Ames. Mountain View, CA: Computer Sciences Corporation.

System Simulation Team. (1987). HAC equations software. Aberdeen Proving Ground, MD: U.S. Army Human Engineering Laboratory.

APPENDIX A

INCLUDE FILES SOURCE CODE

```
C*********************************************************************
C                   S I G N I F I C A N T   V A R I A B L E S
C*********************************************************************
C
C
C 3        12      20                                          63
C VARIABLF COMMON  ---------- D E F I N I T I O N ----------   UNITS
C NAME     ARRAY
C          LOCATION
C
C
C                  > > > >   I N P U T   < < < <
C                  ------------------------------
C
C
C
C SPHI      A( 10)  SINE OF EULER ROLL ANGLE                    NONE
C CPHI      A( 11)  COSINE OF EULER ROLL ANGLE                  NONE
C STHT      A( 12)  SINE OF EULER PITCH ANGLE                   NONE
C CTHT      A( 13)  COSINE OF EULER PITCH ANGLE                 NONE
C SPSI      A( 14)  SINE OF EULER YAW ANGLE                     NONE
C CPSI      A( 15)  COSINE OF EULER YAW ANGLE                   NONE
C T11       A( 16)  CTHT*CPSI                                   NONE
C T21       A( 17)  SPHI*STHT*CPSI - CPHI*SPSI                  NONE
C T31       A( 18)  CPHI*STHT*CPSI + SPHI*SPSI                  NONE
C T12       A( 19)  CTHT*SPSI                                   NONE
C T22       A( 20)  SPHI*STHT*SPSI + CPHI*CPSI                  NONE
C T32       A( 21)  CPHI*STHT*SPSI - SPHI*CPSI                  NONE
C T13       A( 22)  - STHT                                      NONE
C T23       A(  23) SPHI*CTHT                                   NONE
C T33       A(  24) CPHI*CTHT                                   NONE
C PB        A(  37) AIRCRAFT ROLL VELOCITY, B-FRAME              RAD/S
C QB        A( 38)  AIRCRAFT PITCH VELOCITY, B-FRAME            RAD/S
C RB        A( 39)  AIRCRAFT YAW VELOCITY, B-FRAME              RAD/S
C PLB       A( 43)  INSTANTANEOUS ROLL RATE, B-FRAME            RAD/S
C QLB       A( 44)  INSTANTANEOUS PITCH RATE, B-FRAME           RAD/S
C RLB       A( 45)  INSTANTANEOUS YAW RATE, B-FRAME             RAD/S
C PTURB     A( 52)  TURB. AND EFFECTS CONTRIB. TO ROLL R.       RAD/S
C QTURB     A( 53)  TURB. AND EFFECTS CONTRIB. TO PITCH R.      RAD/S
C RTURB     A( 54)  TURB. AND EFFECTS CONTRIB. TO YAW R.        RAD/S
C VN        A( 64)  NORTHWARD VEL. OVER EARTH'S SUR., L-FRAME   F/S
C VE        A( 65)  EASTWARD VEL. OVER FIXED EARTH, L-FRAME     F/S
C VD        A( 66)  DOWNWARD VEL. TOWARD EARTH'S CEN., L-FRAME  F/S
C VNW       A( 76)  NORTH COMPONENT OF WIND VELOCITY (RMS)      F/S
C VEW       A( 77)  EAST COMPONENT OF WIND VELOCITY (RMS)       F/S
C VDW       A( 78)  DOWNWARD COMPONENT OF WIND VELOCITY (RMS)   F/S
C RR        A(108)  RADIUS OF EARTH + HEIGHT OF RUNWAY          FT
C XLATR     A(111)  LATITUDE OF THE RUNWAY                      RAD
C XLONR     A(112)  LONGITUDE OF THE RUNWAY                     RAD
C CLATR     A(113)  COSINE OF THE RUNWAY LATITUDE               NONE
C STHETR    A(114)  SINE OF RUNWAY ANGLE                        NONE
C CTHETR    A(115)  COSINE OF RUNWAY ANGLE                      NONE
```

```
C XMC(1)     A(120)  ((XIYY-XIZZ)*XIZZ-XIXZ**2)/(XIXX*XIZZ*XIXZ**2) NONE
C XMC(2)     A(121)  (XIXX-XIYY+XIZZ)*XIXZ/(XIXX*XIZZ-XIXZ**2)  NONE
C XMC(3)     A(122)  XIZZ/(XIXX*XIZZ-XIXZ**2)                   NONE
C XMC(4)     A(123)  XIXZ/(XIXX*XIZZ-XIXZ**2)                   NONE
C XMC(5)     A(124)  (XIZZ-XIXX)/XIYY                           NONE
C XMC(6)     A(125)  XIXZ/XIYY                                  NONE
C XMC(7)     A(126)  1./XIYY                                    NONE
C XMC(8)     A(127)  ((XIXX-XIYY)*XIXX*XIXZ**2)/(XIXX*XIZZ-XIXZ**2) NONE
C XMC(9)     A(128)  (XIYY-XIZZ-XIXX)*XIXZ/(XIXX*XIZZ-XIXZ**2)  NONE
C XMC(10)    A(129)  XIXX/(XIXX*XIZZ-XIXZ**2)                   NONE
C XMASS      A(130)  MASS OF VEHICLE (INCLUDING FUEL)           SLUGS
C FAX        A(136)  AERODYNAMIC X-FORCE                        LBS
C FAY        A(137)  AERODYNAMIC Y-FORCE                        LBS
C FAZ        A(138)  AERODYNAMIC Z-FORCE                        LBS
C FEX        A(139)  ENGINES X-FORCE                            LBS
C FEY        A(140)  ENGINES Y-FORCE                            LBS
C FEZ        A(141)  ENGINES Z-FORCE                            LBS
C FGX        A(142)  LANDING GEARS X-FORCE                      LBS
C FGY        A(143)  LANDING GEARS Y-FORCE                      LBS
C FGZ        A(144)  LANDING GEARS Z-FORCE                      LBS
C TAL        A(155)  AERODYNAMIC ROLLING MOMENT COMPONENT       F-LBS
C TAM        A(156)  AERODYNAMIC PITCHING MOMENT COMPONENT      F-LBS
C TAN        A(157)  AERODYNAMIC YAWING MOMENT COMPONENT        F-LBS
C TEL        A(158)  ENGINE ROLLING MOMENT COMPONENT            F-LBS
C TEM        A(159)  ENGINE PITCHING MOMENT COMPONENT           F-LBS
C TEN        A(160)  ENGINE YAWING MOMENT COMPONENT             F-LBS
C TGL        A(161)  LANDING GEARS ROLLING MOMENT COMPONENT     F-LBS
C TGM        A(162)  LANDING GEARS PITCHING MOMENT COMPONENT    F-LBS
C TGN        A(163)  LANDING GEARS YAWING MOMENT COMPONENT      F-LBS
C DT2        A(168)  SECOND LOOP TIME FRAME (MEDIUM)            S
C HR         A(170)  HEIGHT OF RUNWAY W/R/T SEA LEVEL           FT
C XP         A(171)  X POSITION OF PILOT W/R/T C.G.             FT
C YP         A(172)  Y POSITION OF PILOT W/R/T C.G.             FT
C ZP         A(173)  Z POSITION OF PILOT W/R/T C.G.             FT
C WAIT       A(177)  AIRCRAFT GROSS WEIGHT                      LB
C SOUNDZ     A(293)  SPEED OF SOUND AT SEA LEVEL                F/S
C RE         A(336)  RADIUS OF EARTH                            FT
C REINV      A(337)  INVERSE OF RE                              1/F
C OMEG       A(347)  EARTH'S ROTATION RATE                      RAD/S
C R2D        A(359)  57.2957795                                 DEG/RAD
C DELAT      A(370)  DELTA AMBIENT TEMPERATURE                  DEG K
C XMCC1      A(375)  XMC4*EXMX - XMC3*EXMX  (ROTATING ENGINES)  1/S
C XMCC2      A(376)  XMC7*EXMX              (ROTATING ENGINES)  1/S
C XMCC3      A(377)  XMC7*EXMZ              (ROTATING ENGINES)  1/S
C XMCC4      A(378)  XMC10*EXMX  - XMC4*EXMZ(ROTATING ENGINES)  1/S
C XMCC5      A(379)  XMC3*EXMY              (ROTATING ENGINES)  1/S
C XMCC6      A(380)  XMC4*EXMY              (ROTATING ENGINES)  1/S
C XMCC7      A(381)  XMC10*EXMY             (ROTATING ENGINES)  1/S
C TLAT       A(426)  TANGENT OF AIRCRAFT LATITUDE               NONE
C
C IMODE      IA(  1) MODE CONTROL INTEGER (NEG:IC, 0:HLD, POS:OP)
C IPLAT      IA(  6) FLAT EARTH OPTION
C IFFCI      IA(  7) DISABLE TRANSLATIONAL DEGREES OF FREEDOM
C IETURB     IA(185) TURB. GENERATED IN EARTH(1) OR BODY(0) FRAME
```

```
C ITOMTR    IA(187) ZERO ALFD AND BETD IN IC
C IFREEZ    IA(200) FREEZE TRANSLATIONAL & ROTATIONAL MOTION OF A.C.
C
C
C
C                   > > > >    O U T P U T    < < < <
C                   ------------------------------
C
C
C
C PHI       A(  1)  ROLL EULER ANGLE,  L-FRAME                   DEG
C THET      A(  2)  PITCH EULER ANGLE,  L-FRAME                  DEG
C PSI       A(  3)  YAW EULER ANGLE,  L-FRAME                    DEG
C PHIR      A(  4)  ROLL ANGLE,  L-FRAME                         RAD
C THETR     A(  5)  PITCH ANGLE,  L-FRAME                        RAD
C PSIR      A(  6)  YAW ANGLE,  L-FRAME                          RAD
C PHID      A(  7)  ROLL RATE,  L-FRAME                          RAD/S
C THED      A(  8)  PITCH RATE,  L-FRAME                         RAD/S
C PSID      A(  9)  YAW RATE,  L-FRAME                           RAD/S
C ALFA      A( 25)  ANGLE OF ATTACK                              DEG
C BETA      A( 26)  SIDESLIP ANGLE                               DEG
C ALFAR     A( 27)  ANGLE OF ATTACK                              RAD
C BETAR     A( 28)  SIDESLIP ANGLE                               RAD
C ALFD      A( 29)  ANGLE OF ATTACK RATE                         RAD/S
C BETD      A( 30)  SIDESLIP ANGLE RATE                          RAD/S
C SALPH     A( 31)  SINE OF ANGLE OF ATTACK                      NONE
C CALPH     A( 32)  COSINE OF ANGLE OF ATTACK                    NONE
C SBETA     A( 33)  SINE OF SIDESLIP ANGLE                       NONE
C CBETA     A( 34)  COSINE OF SIDESLIP ANGLE                     NONE
C GAMV      A( 35)  FLIGHT PATH ANGLE ABOVE HORIZON              RAD
C GAMH      A( 36)  FLIGHT PATH ANGLE IN HORIZON FRAME           RAD
C PL        A( 40)  INSTANTANEOUS ROLL RATE, L-FRAME             RAD/S
C QL        A( 41)  INSTANTANEOUS PITCH RATE, L-FRAME            RAD/S
C RL        A( 42)  INSTANTANEOUS YAW RATE, L-FRAME              RAD/S
C PT        A( 46)  TOTAL ROLL RATE, L-FRAME                     RAD/S
C QT        A( 47)  TOTAL PITCH RATE, L-FRAME                    RAD/S
C RT        A( 48)  TOTAL YAW RATE, L-FRAME                      RAD/S
C PBWN      A( 49)  ROLL RATE, B-FRAME                           RAD/S
C QBWN      A( 50)  PITCH RATE, B-FRAME                          RAD/S
C RBWN      A( 51)  YAW RATE, B-FRAME                            RAD/S
C PBD       A( 55)  AIRCRAFT ROLL ACCELERATION, B-FRAME          RAD/S2
C QBD       A( 56)  AIRCRAFT PITCH ACCELERATION, B-FRAME         RAD/S2
C RBD       A( 57)  AIRCRAFT YAW ACCELERATION, B-FRAME           RAD/S2
C UB        A( 58)  X VELOCITY (FORWARD), B-FRAME                F/S
C VB        A( 59)  Y VELOCITY (RIGHTWARD), B-FRAME              F/S
C WB        A( 60)  Z VELOCITY (DOWNWARD), B-FRAME               F/S
C UTURB     A( 61)  X VELOCITY TURBULENCE COMPONENT, B-FRAME     F/S
C VTURB     A( 62)  Y VELOCITY TURBULENCE COMPONENT, B-FRAME     F/S
C WTURB     A( 63)  Z VELOCITY TURBULENCE COMPONENT              F/S
C VEE       A( 67)  EASTWARD VELOCITY OVER ROTATING EARTH        F/S
C VT        A( 68)  TOTAL VELOCITY, L-FRAME                      F/S
C VG        A( 69)  GROUND SPEED, L-FRAME                        F/S
C VRW       A( 70)  VELOCITY W/R/T WIND                          F/S
C VNR       A( 72)  NORTH RELATIVE VELOCITY                      F/S
```

```
C VER       A( 73)   EASTWARD RELATIVE VELOCITY                     F/S
C VDR       A( 74)   DOWNWARD RELATIVE VELOCITY                     F/S
C VEQ       A( 75)   EQUIVALENT AIRSPEED                            KNOTS
C ALTD      A( 80)   ALTITUDE RATE OF CHANGE                        F/S
C XLOND     A( 81)   LONGITUDE RATE OF CHANGE                       RAD/S
C XLATD     A( 82)   LATITUDE RATE OF CHANGE                        RAD/S
C ALT       A( 83)   ALTITUDE OF AIRCRAFT ABOVE SEA LEVEL           F
C XLON      A( 84)   LONGITUDE OF AIRCRAFT                          RAD
C XLAT      A( 85)   LATITUDE OF AIRCRAFT                           RAD
C SLAT      A( 86)   SINE OF AIRCRAFT LATITUDE                      NONE
C CLAT      A( 87)   COSINE OF AIRCRAFT LATITUDE                    NONE
C VND       A( 88)   NORTHWARD ACCELERATION OVER EARTH, L-FRAME     F/S2
C VED       A( 89)   EASTWARD ACCELERATION OVER EARTH, L-FRAME      F/S2
C VDD       A( 90)   DOWNWARD ACCELERATION TOWARD EARTH CENTER      F/S2
C AX        A( 91)   X ACCELERATION OF AIRCRAFT C.G., B-FRAME       F/S2
C AY        A( 92)   Y ACCELERATION OF AIRCRAFT C.G., B-FRAME       F/S2
C AZ        A( 93)   Z ACCELERATION OF AIRCRAFT C.G., B-FRAME       F/S2
C AXP       A( 94)   X ACCELERATION OF PILOT, B-FRAME               F/S2
C AYP       A( 95)   Y ACCELERATION OF PILOT, B-FRAME               F/S2
C AZP       A( 96)   Z ACCELERATION OF PILOT, B-FRAME               F/S2
C G         A( 97)   ACCERATION DUE TO GRAVITY (AT ALTITUDE)        F/S2
C VCAL      A(101)   CALIBRATED AIRSPEED                            KNOTS
C XPR       A(103)   DISTANCE OF PILOT DOWN RUNWAY                  F
C YPR       A(104)   DISTANCE OF PILOT TO THE RIGHT OF RUNWAY       F
C HPR       A(105)   HEIGHT OF PILOT ABOVE RUNWAY                   F
C DNR       A(106)   AIRCRAFT DISTANCE NORTH OF RUNWAY THRESHOL     F
C DER       A(107)   AIRCRAFT DISTANCE EAST OF RUNWAY THRESHOLD     F
C RTV       A(109)   RADIUS FROM EARTH'S CENTER TO AIRCRAFT         F
C FTX       A(145)   TOTAL AIRCRAFT X FORCE (FORWARD)               LBS
C FTY       A(146)   TOTAL AIRCRAFT Y FORCE (RIGHTWARD)             LBS
C FTZ       A(147)   TOTAL AIRCRAFT Z FORCE (DOWNWARD)              LBS
C FN        A(148)   TOTAL FORCE NORTH, L-FRAME                     LBS
C FE        A(149)   TOTAL FORCE EAST, L-FRAME                      LBS
C FD        A(150)   TOTAL FORCE DOWN TOWARD EARTH CENTER           LBS
C FG        A(151)   FORCE DUE TO GRAVITY (AT ALT.) L-FRAME         LBS
C TTL       A(164)   TOTAL AIRCRAFT ROLL MOMENT                     F-LBS
C TTM       A(165)   TOTAL AIRCRAFT PITCH MOMENT                    F-LBS
C TTN       A(166)   TOTAL AIRCRAFT YAW MOMENT                      F-LBS
C XCG       A(174)   C.G. X POSITION W/R/T RUNWAY FRAME             F
C YCG       A(175)   C.G. Y POSITION W/R/T RUNWAY FRAME             F
C HCG       A(176)   C.G. HEIGHT ABOVE RUNWAY                       F
C QBAR      A(178)   DYNAMIC PRESSURE                               LB/F2
C CBARC     A(179)   IMPACT PRESSURE                                LB/F2
C RHO       A(183)   AIR DENSITY AT ALTITUDE                        SLUG/F3
C SOUND     A(209)   VELOCITY OF SOUND AT ALTITUDE                  F/S
C TIME      A(303)   TIME SINCE START OF OPERATE MODE               SEC
C TR        A(332)   TOTAL TEMPERATURE RATIO                        NONE
C PR        A(333)   TOTAL PRESSURE RATIO                           NONE
C RHOZ      A(364)   SEA LEVEL DENSITY (STANDARD DAY)               SLUG/F3
C TAMB      A(366)   AMBIENT TEMPERATURE                            DEG K
C PAMB      A(367)   AMBIENT PRESSURE                               LB/F2
C TTOT      A(368)   TOTAL TEMPERATURE                              DEG K
C PTOT      A(369)   TOTAL PRESSURE                                 LB/F2
C UBD       A(413)   FORWARD ACCELERATION, B-FRAME                  F/S2
```

```
C VBD       A(414)  RIGHTWARD ACCELERATION, B-FRAME            F/S2
C WBD       A(415)  DOWNWARD ACCELERATION, B-FRAME             F/S2
C VTWN      A(416)  TOTAL WIND VEL.(NORTH) INCL. TURB.         F/S
C VTWE      A(417)  TOTAL WIND VEL.(EAST)  INCL. TURB.         F/S
C VTWD      A(418)  TOTAL WIND VEL.(DOWN)  INCL. TURB.         F/S
C VNTURB    A(419)  RANDOM COMPONENT (NORTH) TURB. VELOCITY    F/S
C VETURB    A(420)  RANDOM COMPONENT (EAST)  TURB. VELOCITY    F/S
C VDTURB    A(421)  RANDOM COMPONENT (DOWN)  TURB. VELOCITY    F/S
C PAMBR     A(424)  RATIO OF AMB. PRESSURE TO SEA LEV. PRES.   NONE
C TAMBR     A(425)  RATIO OF AMB. TEMP. TO SEA LEV. TEMP.      NONE
C
C ICOND     IA(141) CONSTANT DENSITY SELECTION SWITCH
C
C
C
C*********************************************************************
C                  C O M M O N S
C*********************************************************************
C
C
      COMMON/FARMCOP/ARM(400)
      COMMON /IAAHCM/IAAH(50)
      COMMON /ICNTRL/ICOZERO! ADDED 6/26/87 GLH
      COMMON /IFIXED/ IA(250)
      COMMON /IRMCOP/IRM(60)
      COMMON/XFLOAT/A(500)

C
C
C*********************************************************************
C                  E Q U I V A L E N C I E S
C*********************************************************************
C
      EQUIVALENCE (A(  1), PHI)
      EQUIVALENCE (A(  2), THET)
      EQUIVALENCE (A(  3), PSI)
      EQUIVALENCE (A(  4), PHIR)
      EQUIVALENCE (A(  5), THETR)
      EQUIVALENCE (A(  6), PSIR)
      EQUIVALENCE (A(  7), PHID)
      EQUIVALENCE (A(  8), THED)
      EQUIVALENCE (A(  9), PSID)
      EQUIVALENCE (A( 10), SPHI)
      EQUIVALENCE (A( 11), CPHI)
      EQUIVALENCE (A( 12), STHT)
      EQUIVALENCE (A( 13), CTHT)
      EQUIVALENCE (A( 14), SPSI)
      EQUIVALENCE (A( 15), CPSI)
      EQUIVALENCE (A( 16), T11)
      EQUIVALENCE (A( 17), T21)
      EQUIVALENCE (A( 18), T31)
      EQUIVALENCE (A( 19), T12)
      EQUIVALENCE (A( 20), T22)
      EQUIVALENCE (A( 21), T32)
```

```
EQUIVALENCE (A( 22), T13)
EQUIVALENCE (A( 23), T23)
EQUIVALENCE (A( 24), T33)
EQUIVALENCE (A( 25), ALFA)
EQUIVALENCE (A( 26), BETA)
EQUIVALENCE (A( 27), ALFAR)
EQUIVALENCE (A( 28), BETAR)
EQUIVALENCE (A( 29), ALFD)
EQUIVALENCE (A( 30), BETD)
EQUIVALENCE (A( 31), SALPH)
EQUIVALENCE (A( 32), CALPH)
EQUIVALENCE (A( 33), SBETA)
EQUIVALENCE (A( 34), CBETA)
EQUIVALENCE (A( 35), GAMV)
EQUIVALENCE (A( 36), GAMH)
EQUIVALENCE (A( 37), PB)
EQUIVALENCE (A( 38), QB)
EQUIVALENCE (A( 39), RB)
EQUIVALENCE (A( 40), PL)
EQUIVALENCE (A( 41), QL)
EQUIVALENCE (A( 42), RL)
EQUIVALENCE (A( 43), PLB)
EQUIVALENCE (A( 44), QLB)
EQUIVALENCE (A( 45), RLB)
EQUIVALENCE (A( 46), PT)
EQUIVALENCE (A( 47), QT)
EQUIVALENCE (A( 48), RT)
EQUIVALENCE (A( 49), PBWN)
EQUIVALENCE (A( 50), QBWN)
EQUIVALENCE (A( 51), RBWN)
EQUIVALENCE (A( 52), PTURB)
EQUIVALENCE (A( 53), QTURB)
EQUIVALENCE (A( 54), RTURB)
EQUIVALENCE (A( 55), PBD)
EQUIVALENCE (A( 56), QBD)
EQUIVALENCE (A( 57), RBD)
EQUIVALENCE (A( 58), UB)
EQUIVALENCE (A( 59), VB)
EQUIVALENCE (A( 60), WB)
EQUIVALENCE (A( 61), UTURB)
EQUIVALENCE (A( 62), VTURB)
EQUIVALENCE (A( 63), WTURB)
EQUIVALENCE (A( 64), VN)
EQUIVALENCE (A( 65), VE)
EQUIVALENCE (A( 66), VD)
EQUIVALENCE (A( 67), VEE)
EQUIVALENCE (A( 68), VT)
EQUIVALENCE (A( 69), VG)
EQUIVALENCE (A( 70), VRW)
EQUIVALENCE (A( 71), XMACH)
EQUIVALENCE (A( 72), VNR)
EQUIVALENCE (A( 73), VER)
EQUIVALENCE (A( 74), VDR)
EQUIVALENCE (A( 75), VEQ)
```

```
EQUIVALENCE (A( 76), VNW)
EQUIVALENCE (A( 77), VEW)
EQUIVALENCE (A( 78), VDW)
EQUIVALENCE (A( 80), ALTD)
EQUIVALENCE (A( 81), XLOND)
EQUIVALENCE (A( 82), XLATD)
EQUIVALENCE (A( 83), ALT)
EQUIVALENCE (A( 84), XLON)
EQUIVALENCE (A( 85), XLAT)
EQUIVALENCE (A( 86), SLAT)
EQUIVALENCE (A( 87), CLAT)
EQUIVALENCE (A( 88), VND)
EQUIVALENCE (A( 89), VED)
EQUIVALENCE (A( 90), VDD)
EQUIVALENCE (A( 91), AX)
EQUIVALENCE (A( 92), AY)
EQUIVALENCE (A( 93), AZ)
EQUIVALENCE (A( 94), AXP)
EQUIVALENCE (A( 95), AYP)
EQUIVALENCE (A( 96), AZP)
EQUIVALENCE (A(97), G)
EQUIVALENCE (A(101), VCAL)
EQUIVALENCE (A(103), XPR)
EQUIVALENCE (A(104), YPR)
EQUIVALENCE (A(105), HPR)
EQUIVALENCE (A(106), DNR)
EQUIVALENCE (A(107), DER)
EQUIVALENCE (A(108), RR)
EQUIVALENCE (A(109), RTV)
EQUIVALENCE (A(111), XLATR)
EQUIVALENCE (A(112), XLONR)
EQUIVALENCE (A(113), CLATR)
EQUIVALENCE (A(114), STHETR)
EQUIVALENCE (A(115), CTHETR)
EQUIVALENCE (A(130), XMASS)
EQUIVALENCE (A(136), FAX)
EQUIVALENCE (A(137), FAY)
EQUIVALENCE (A(138), FAZ)
EQUIVALENCE (A(139), FEX)
EQUIVALENCE (A(140), FEY)
EQUIVALENCE (A(141), FEZ)
EQUIVALENCE (A(142), FGX)
EQUIVALENCE (A(143), FGY)
EQUIVALENCE (A(144), FGZ)
EQUIVALENCE (A(145), FTX)
EQUIVALENCE (A(146), FTY)
EQUIVALENCE (A(147), FTZ)
EQUIVALENCE (A(148), FN)
EQUIVALENCE (A(149), FE)
EQUIVALENCE (A(150), FD)
EQUIVALENCE (A(151), FG)
EQUIVALENCE (A(155), TAL)
EQUIVALENCE (A(156), TAM)
EQUIVALENCE (A(157), TAN)
```

```
EQUIVALENCE (A(158), TEL)
EQUIVALENCE (A(159), TEM)
EQUIVALENCE (A(160), TEN)
EQUIVALENCE (A(161), TGL)
EQUIVALENCE (A(162), TGM)
EQUIVALENCE (A(163), TGN)
EQUIVALENCE (A(164), TTL)
EQUIVALENCE (A(165), TTM)
EQUIVALENCE (A(166), TTN)
EQUIVALENCE (A(168), DT2 )
EQUIVALENCE (A(170), HR)
EQUIVALENCE (A(171), XP)
EQUIVALENCE (A(172), YP)
EQUIVALENCE (A(173), ZP)
EQUIVALENCE (A(174), XCG)
EQUIVALENCE (A(175), YCG)
EQUIVALENCE (A(176), HCG)
EQUIVALENCE (A(177), WAIT)
EQUIVALENCE (A(178), QBAR)
EQUIVALENCE (A(179), QBARC)
EQUIVALENCE (A(183), RHO)
EQUIVALENCE (A(209), SOUND)
EQUIVALENCE (A(231), THETIC)
EQUIVALENCE (A(232), PSIIC)

EQUIVALENCE (A(293), SOUNDZ)
EQUIVALENCE (A(303), TIME)
EQUIVALENCE (A(332), TR)
EQUIVALENCE (A(333), PR)
EQUIVALENCE (A(336), RE)
EQUIVALENCE (A(337), REINV)
EQUIVALENCE (A(347), OMEG)
EQUIVALENCE (A(359), R2D)
EQUIVALENCE (A(358), D2R)

EQUIVALENCE (A(365), HRHOZ)
EQUIVALENCE (A(366), TAMB)
EQUIVALENCE (A(367), PAMB)
EQUIVALENCE (A(368), TTOT)
EQUIVALENCE (A(369), PTOT)
EQUIVALENCE (A(370), DELAT)
EQUIVALENCE (A(375), XMCC1)
EQUIVALENCE (A(376), XMCC2)
EQUIVALENCE (A(377), XMCC3)
EQUIVALENCE (A(378), XMCC4)
EQUIVALENCE (A(379), XMCC5)
EQUIVALENCE (A(380), XMCC6)
EQUIVALENCE (A(381), XMCC7)
EQUIVALENCE (A(413), UBD)
EQUIVALENCE (A(414), VBD)
EQUIVALENCE (A(415), WBD)
EQUIVALENCE (A(416), VTWN)
EQUIVALENCE (A(417), VTWE)
EQUIVALENCE (A(418), VTWD)
```

```
      EQUIVALENCE (A(419), VNTURB)
      EQUIVALENCE (A(420), VETURB)
      EQUIVALENCE (A(421), VDTURB)
      EQUIVALENCE (A(424), PAMBR)
      EQUIVALENCE (A(425), TAMBR)
      EQUIVALENCE (A(426), TLAT)
C
      EQUIVALENCE (IA(  1), IMODE)
      EQUIVALENCE (IA(  6), IFLAT)
      EQUIVALENCE (IA(  7), IFFCI)
      EQUIVALENCE (IA( 29), IAND)

      EQUIVALENCE (IA( 33), ILWD)

      EQUIVALENCE (IA(141), ICOND)
      EQUIVALENCE (IA(185), IETURB)
      EQUIVALENCE (IA(187), ITOMTR)
      EQUIVALENCE (IA(200), IFREEZ)
C
      EQUIVALENCE (ZLVT    ,XHO     (  1))
      EQUIVALENCE (ZMUT    ,XHO     (  2))
      EQUIVALENCE (ZMWT    ,XHO     (  3))
      EQUIVALENCE (ZNVT    ,XHO     (  4))
      EQUIVALENCE (ZUT     ,XHO     (  5))
      EQUIVALENCE (ZWTT    ,XHO     (  6))
      EQUIVALENCE (ZLPT    ,XHO     (  7))
      EQUIVALENCE (XLPHIO  ,XHO     (  8))
      EQUIVALENCE (XLDAT   ,XHO     (  9))
      EQUIVALENCE (ZMQT    ,XHO     ( 10))
      EQUIVALENCE (XMTHETO ,XHO     ( 11))
      EQUIVALENCE (XMDET   ,XHO     ( 12))
      EQUIVALENCE (ZNRT    ,XHO     ( 13))
      EQUIVALENCE (XNDPT   ,XHO     ( 14))
      EQUIVALENCE (ZWT     ,XHO     ( 15))
      EQUIVALENCE (ZDCT    ,XHO     ( 16))
      EQUIVALENCE (XUT     ,XHO     ( 17))
      EQUIVALENCE (YVT     ,XHO     ( 18))
      EQUIVALENCE (ZMU     ,XHO     ( 19))
      EQUIVALENCE (ZLV     ,XHO     ( 20))
      EQUIVALENCE (FPS2KOS ,XHO     ( 23))
      EQUIVALENCE (XNVO    ,XHO     ( 24))
      EQUIVALENCE (ZNPO    ,XHO     ( 25))
      EQUIVALENCE (XNPHIO  ,XHO     ( 26))
      EQUIVALENCE (XNV     ,XHO     ( 27))
      EQUIVALENCE (ZUO     ,XHO     ( 28))
      EQUIVALENCE (ROLLOP  ,XHO     ( 90))
      EQUIVALENCE (PITCHOP ,XHO     ( 91))
      EQUIVALENCE (YAWOP   ,XHO     ( 92))
      EQUIVALENCE (COLOP   ,XHO     ( 93))

      EQUIVALENCE ( ARM    (  4)  ,  COLTRM      )
      EQUIVALENCE ( ARM    (  5)  ,  FRX         )
      EQUIVALENCE ( ARM    (  6)  ,  FRY         )
      EQUIVALENCE ( ARM    (  7)  ,  FRZ         )
```

```
      EQUIVALENCE ( ARM     (  8)   ,  PBDR         )
      EQUIVALENCE ( ARM     (  9)   ,  QBDR         )
      EQUIVALENCE ( ARM     ( 10)   ,  RBDR         )
      EQUIVALENCE ( ARM     ( 23)   ,  SGVOL        )
      EQUIVALENCE ( ARM     ( 24)   ,  SGVOLIC      )
      EQUIVALENCE ( ARM     ( 25)   ,  SGVOLOP      )
      EQUIVALENCE ( ARM     ( 26)   ,  TINC         )
      EQUIVALENCE ( ARM     ( 31)   ,  TINCIC       )
      EQUIVALENCE ( ARM     ( 32)   ,  TINCOP       )
      EQUIVALENCE ( ARM     ( 33)   ,  ROLGRD       )
      EQUIVALENCE ( ARM     ( 34)   ,  ROLBRK       )
      EQUIVALENCE ( ARM     ( 35)   ,  ROLHYS       )
      EQUIVALENCE ( ARM     ( 36)   ,  PCHGRD       )
      EQUIVALENCE ( ARM     ( 37)   ,  PCHBRK       )
      EQUIVALENCE ( ARM     ( 38)   ,  COLHYS       )
      EQUIVALENCE ( ARM     ( 39)   ,  PEDGRD       )
      EQUIVALENCE ( ARM     ( 40)   ,  PEDBRK       )
      EQUIVALENCE ( ARM     ( 41)   ,  PEDHYS       )
      EQUIVALENCE ( ARM     ( 42)   ,  CLCFRC       )
      EQUIVALENCE ( ARM     ( 43)   ,  CLCBRK       )
      EQUIVALENCE ( ARM     ( 51)   ,  TRIM1C       )
      EQUIVALENCE ( ARM     ( 52)   ,  TRIM2C       )
      EQUIVALENCE ( ARM     ( 53)   ,  TRIM3C       )
      EQUIVALENCE ( ARM     ( 54)   ,  CLCGRD       )
      EQUIVALENCE ( ARM     (108)   ,  ANZ          )
      EQUIVALENCE ( ARM     (110)   ,  BETAHUD      )
      EQUIVALENCE ( ARM     (113)   ,  PHIRDM       )
      EQUIVALENCE ( ARM     (209)   ,  QRAPCT       )
      EQUIVALENCE ( ARM     (363)   ,  FIMB         )
C
      EQUIVALENCE (ICON     ,IAAH      (15),ICONFI)
      EQUIVALENCE (ICS      ,IAAH      (20)       )
      EQUIVALENCE (MCS      ,IAAH      (21)       )
      EQUIVALENCE (MCON     ,IAAH      (22)       )
C
      EQUIVALENCE ( IRM     ( 48)   ,  LIMG         )
      EQUIVALENCE ( IRM     ( 49)   ,  LIMRPM       )
      EQUIVALENCE ( IRM     ( 50)   ,  LIMBET       )
      EQUIVALENCE ( IRM     ( 51)   ,  BETAFG       )
```

HAC2.INC

```
C*************************************************************
C                          HAC2.INC
C*************************************************************
      COMMON/HACDAT/ROLLO,PITCHO,YAWO,ROLMAX,PITMAX,YAWMAX,COLOMIN,
     & COLOMAX,RDBO,PDBO,YDBO,CDBO,OOOG1,OOOG2,OOOG3,
     & OOOG4,OOOB4,IMBC,ILITE,GNUD,GRLD,TRIM1,TRIM1P,TRIM2,
     & TRIM2P,TRIM3,TRIM3P,TRIM4,TRIM4C,TRIM4P,COLMINT,
     & COLMAXT,PCTBIAS,CBAUTO,CGAUTO,CFMAN,CFAUTO,
     & XRT,YRT,ZRT,XKDP,XLPHI1,XLPHI2,XLPHI3,
     & ZLPD,ZMQD,ZMQ1,ZMQ2,ZLP1,ZLP2,XLDA1,XLDA3,XMDE1,
     & XMDE3,SLP1,YINT1,SLP2,YINT2,SLP3,YINT3,SLP4,YINT4,
     & SLP5,YINT5,RODSLP,RODINT1,RODINT2,THETV1C,
     & THETOFF,TAUCOL,GCOL,APPPCT,REDTIME,LTRIM,
     & COLOZ,COLF1,COLF2,ROCLIM,COLO,EROM4,
     & HLEV,XM1,PRESZ,TEMPZ,YCON,TEMA,DELT,DELTH,DELTI,
     & ESLP1(3),ESLP2(3),EINT1(3),EINT2(3),VEQPT(3),
     & RSLP1(3),RSLP2(3),XINT1(3),XINT2(3),
     & VQMAX(6),RTMAX(6),VQMIN(6),RTMIN(6),
     & XMC(10),YMINT(6),RYINT(6),SLPMN(6),SLPMX(6),
     & AAP(2),BBP(3),BUFFP(7),XXP(2),AAR(2),BBR(3),
     & BUFFR(7),XXR(2)
```

```
C*****************************************************************
C                              HAC3.INC
C*****************************************************************
 LOGICAL*4  alt_hold    !Freeze altitude.  MCL 01/22/88
 LOGICAL*4  airspeed_hold     !Freeze airspeed.  MCL 01/25/88
 INTEGER*4  chad(6)     !Contains the a/d inputs.
 LOGICAL*4  clready     !True when collective is trimmed
 REAL*4     clstk !Delta on collective when mode is 4
 REAL*4     coloffset
 REAL*4     colscalar
 REAL*4     coloptol    !Primary control stick setting for !collective
 LOGICAL*4  heading_hold      !Freeze heading.  MCL 01/25/88
 REAL*4     pitchoptol  !Primary control stick setting for !pitch
 REAL*4     rolloptol   !Primary control stick setting for roll
 LOGICAL*4  gocom !Set to true by the pilot to start !flying
 LOGICAL*4  lclose7     !True when file 7 (Vectrix) was closed
 INTEGER*4  lpamode     !Used on main.for
 INTEGER*4  mode        !Mode of fly
 REAL*4     pedoffset
 REAL*4     pedscalar
 REAL*4     pitoffset
 REAL*4     pitscalar
 REAL*4     pstk        !Delta on pitch when mode is 4
 LOGICAL*4  ptready     !True when pitch is trimmed
 LOGICAL*4  rdready     !True when pedals are trimmed
 REAL*4     rdstk !Delta on pedals when mode is 4
 LOGICAL*4  rlready     !True when roll is trimmed
 REAL*4     rlstk !Delta on roll when mode is 4
 REAL*4     roloffset
 REAL*4     rolscalar
 REAL*4     yawoptol    !Primary control stick setting for !pedals
 INTEGER*4  iallmulax   !When set to 1 will provide collective !and yaw on
                              multiaxis controller.
 COMMON/HACTRIM/
     & chad,lclose7,clready,clstk,coloffset,colscalar,
     & coloptol,yawoptol,pitchoptol, alt_hold, airspeed_hold,
     & heading_hold, rolloptol,gocom,lpamode,mode,pedoffset,
     & pedscalar,pitoffset,pitscalar,pstk,ptready,rdready,
     & rdstk,rlready,rlstk,roloffset,rolscalar,iallmulax
```

MOVGBL.INC

```
C************************************************************
c                        MOVGBL.INC
C************************************************************
 INTEGER*4          MOX(10)
 INTEGER*4          MOY(10)
 INTEGER*4          MOZ(10)

 INTEGER*2          ATYPE(10)
 INTEGER*2          ACAT(10)
 INTEGER*2          AGROUP(10)

 REAL*4             MOU(10)
 REAL*4             MOHEAD(10)
 REAL*4             MOPHI(10)
 REAL*4             MOTHET(10)

 COMMON/MOVOBJGBL/MOX,MOY,MOZ,ATYPE,ACAT,AGROUP,
     &  MOU,MOHEAD,MOPHI,MOTHET
```

FLTDATA.INC

```
C*************************************************************
C                         FLTDATA.INC
C*************************************************************
      LOGICAL*1    INIT_DATA_COL,CRASH_DET
      LOGICAL*1    BEGIN_FLT
      LOGICAL*1    START_TALK
      LOGICAL*1    DYNAMIC
      LOGICAL*1    VOICE_BEGIN
      LOGICAL*1    WAYPT_BEGIN(8),WAYPT_SENT(8)
      LOGICAL*1    END_FLIGHT
      LOGICAL*1    WAYPT_READY !mcl 04-01-1986 Tells when
                               !waypoint is ready to be
                               !entered.
      CHARACTER*4  key_input
      LOGICAL*1    BEGIN_ENTER(8)
      INTEGER*2    WAYPT_NUM
      BYTE         WAYPT(10)
      INTEGER*4    WAYPT_ERR,FSX2,FSY2
      INTEGER*4    TEST_MODE
      REAL         TURN_RATE

      COMMON/YFLTDAT/WAYPT_NUM,WAYPT,WAYPT_SENT,WAYPT_BEGIN,
     & INIT_DATA_COL,BEGIN_FLT,CRASH_DET,START_TALK,
     & TEST_MODE,DYNAMIC,VOICE_BEGIN,FSX2,FSY2,key_input,
     & TURN_RATE,WAYPT_READY,BEGIN_ENTER,END_FLIGHT,
     & WAYPT_ERR
```

APPENDIX B

MAIN PROGRAMS SOURCE CODE

START_HACSEC.FOR

```
C*****************************************************************
            Program START_HACSEC          !7 december 1987 mcl
C   Author:  Maria del C. Lopez
C
C    This program creates the global section to be used by the hac equations
c    on CANCER.
c
c   Directory:
c           CANCER::[helhac.for]
c
c   Subroutines called:
c           Get_chan
c
c   Files opened:
c           Hacsec.dat-----------------'unknown'
c
c   Global section name:
c           HACSEC
c
c   Common section name:
c           ACHANNEL
c*****************************************************************
  INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
           INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
  INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
  INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
  INCLUDE '[HELHAC.FOR}FLTDATA.INC/LIST'

  INCLUDE '($SECDEF)'
  INCLUDE '($SSDEF)'
  INCLUDE '($IODEF)'

  INTEGER*4 SYS$CRMPSC,STATUS
  INTEGER*4 SEC_FLAGS,MAPRANGE(2),RETADR(2)
  INTEGER*4 GET_CHAN
  INTEGER*2 SEC_CHAN

  REAL*4    F1

  CHARACTER*6 HAC1SEC/'HACSEC'/

  EXTERNAL GET_CHAN

  COMMON /ACHANNEL/SEC_CHAN
c...cluster
 INTEGER*4  sys$ascefc
 INTEGER*4  sys$waitfr,sys$clref,sys$setef
 CHARACTER*8      clname3 /'CLUSTER3'/

c   Associate cluster 3
 status = sys$ascefc(%val(96),clname3,,)
 if (.not. status) call lib$stop(%val(status))
```

```
C  Create section to share data with DR11 and A/D30Hz processes

   MAPRANGE(1) = %LOC(ARM(1))
   MAPRANGE(2) = %LOC(WAYPT_ERR)

   SEC_FLAGS = SEC$M_GBL .OR. SEC$M_WRT .OR. SEC$M_DZRO

   OPEN(UNIT=1,FILE='HACSEC.DAT',USEROPEN=GET_CHAN,SHARED,
     &         INITIALSIZE=60,STATUS='UNKNOWN')

   STATUS = SYS$CRMPSC(MAPRANGE,RETADR,,%VAL(SEC_FLAGS),
     & HAC1SEC,,,%VAL(SEC_CHAN),%VAL(60),,,)
   IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))


  STATUS = SYS$SUSPND(,)
  IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

  end
```

AD30HZ.FOR

```
      PROGRAM AD30HZ
C A/D values from the ADV11 A/D using memory mapped I/O
C
C This example shows the synchronous user interface (LIO$READ)
C and memory mapped I/O
C
C First, the program sets up the ADV as follows:
C       use A/D channels 0 THRU 15
C       use a gain of one
C       start immediately on the LIO$READ call and keep cycling through the
C       selected channels as fast as the A/D can go until the buffer is full
C
C Second, the program reads the data into the buffer.
C
C
C Compile, link, and run the program as follows:
C       FORTRAN example4_1
C       LINK example4_1
C       RUN example4_1

        INCLUDE 'sys$library:LIOSET.FOR'   !LIO$SET_I symbol definitions

 INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
 INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
 INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
 INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
 INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'

    INCLUDE '($SECDEF)'
 INCLUDE '($SSDEF)'

    INTEGER*2       buffer(17)       !twenty word buffer
    INTEGER         device_ID        !device ID variable
    INTEGER         data_length      !amount of data read (in
                                              !bytes)
    INTEGER         status           !status returned by LIO
                                              !calls
 INTEGER          COUNTER

 INTEGER*4  SYS$SETIMR,SYS$CLREF,SYS$WAITFR
 INTEGER*4  SYS$MGBLSC,SYS$SETEF,SYS$BINTIM
 INTEGER*4  BINARY_INTERVAL(2)
 INTEGER*4  SEC_FLAGS,MAPRANGE(2),RETADR(2)

 CHARACTER  ASCII_INTERVAL*10 /'0 0:0:0.03'/
 CHARACTER*8 CLUSTER_NAME/'CLUSTER3'/
 CHARACTER*6 HAC1SEC/'HACSEC'/
```

```
          COMMON /ADCOM/BUFFER

          COUNTER = 0

          STATUS= SYS$BINTIM(ASCII_INTERVAL,BINARY_INTERVAL)
          IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

          STATUS = SYS$ASCEFC(%VAL(96),CLUSTER_NAME,,)
          IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

          MAPRANGE(1) = %LOC(ARM(1))
          MAPRANGE(2) = %LOC(WAYPT_ERR)

          SEC_FLAGS = SEC$M_WRT

          STATUS = SYS$MGBLSC(MAPRANGE,RETADR,,%VAL(SEC_FLAGS),
     &                    HAC1SEC,,)
          IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))


C Attach to the A/D
C       Gets a device_ID for the ADV and tells LIO to use memory
C       mapped I/O
C
        status = LIO$ATTACH(device_ID, 'AZA0', LIO$K_MAP) !attach
                                                          !to axv
        IF(.NOT.(status)) CALL lib$signal(%val(status))

C Set up the A/D
C       synchronous I/O (LIO$READ/LIO$WRITE)
C       A/D channels zero to FIFTEEN
C       A/D gain of 1 for all five channels
C       Trigger mode = start on LIO$READ and fill buffer as fast
C                      as possible
C
        status = LIO$SET_I(device_ID, LIO$K_SYNCH, 0)
        IF(.NOT.(status)) CALL lib$signal(%val(status))
        status = LIO$SET_I(device_ID, LIO$K_AD_CHAN, 16,
     &           0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
        IF(.NOT.(status)) CALL lib$signal(%val(status))
        status = LIO$SET_I(device_ID, LIO$K_AD_GAIN, 16,
      &           1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
        IF(.NOT.(status)) CALL lib$signal(%val(status))
        status = LIO$SET_I(device_ID, LIO$K_TRIG, 1,
     &           LIO$K_IMM_BURST)
        IF(.NOT.(status)) CALL lib$signal(%val(status))

C SET THE INTERVAL BETWEEN A/D CONVERSIONS
100       STATUS=SYS$SETIMR(%VAL(1),BINARY_INTERVAL,,)
 IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C read 16 A/D values (32 bytes) into the buffer
C       The number of bytes transferred is returned in
```

```
C       data_length
C       The device specific parameter is not used so it is
C       defaulted
C
        status = LIO$READ(device_ID, buffer, 34, data_length, )
        IF(.NOT.(status)) CALL lib$signal(%val(status))

 CHAD(1) = BUFFER(10) ! MULTIAXIS PITCH
 CHAD(2) = BUFFER(11) ! MULTIAXIS ROLL
 CHAD(3) = BUFFER(2)  ! CONVENTIONAL COLLECTIVE
 CHAD(4) = BUFFER(3)  ! CONVENTIONAL YAW
 CHAD(5) = BUFFER(4)  ! MULTIAXIS COLLECTIVE
 CHAD(6) = BUFFER(7)  ! MULTIAXIS YAW
 COUNTER = COUNTER + 1

 STATUS = SYS$SETEF(%VAL(96))
 IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

 IF(COUNTER .EQ. 2)THEN
   STATUS = SYS$SETEF(%VAL(97))
       IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   COUNTER = 0
 ENDIF

C WAIT FOR THE TIMER TO TIME OUT AS SIGNALED BY EVENT FLAG 1.
 STATUS=SYS$WAITFR(%VAL(1))
 IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C GET DATE AND TIME FOR EACH CONVERSION
 STATUS= LIB$DATE_TIME( CUR_TIME)
 IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C CLEAR EVENT FLAG 1 AND PREPARE TO READ A/D AGAIN.
 STATUS=SYS$CLREF(%VAL(1))
 IF(.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C CHECK RATE OF CONVERSION
C           WRITE(10,99)CUR_TIME
C99         FORMAT(A23)


C WRITE DATA TO DISK FILE. NOTE!! CANNOT SUSTAIN 30HZ RATE
C WHEN WRITING TO DISK.
C        WRITE(10,1010)(buffer(i),i=1,16)
C 1010    FORMAT(16(2x,I6))

 GOTO 100

C detach the device
C Rundown is not relevent for synchronous I/O, so it is
C defaulted
C
        status = LIO$DETACH(device_ID, )
        IF(.NOT.(status)) CALL lib$signal(%val(status))
```

```
STOP 'Done'
END
```

DR11_SND_RCV.FOR

```
c*************************************************************
c
c                    PROGRAM DR11_SND_RCV.FOR
c
c The following privilges are required to run the drll software:
c
c                    $ SET PROC/PRIV=PHY_IO
c                    $ SET PROC/PRIV=LOG_IO
c
c The following macro programs must have been compiled:
c
c                    $ MAC XAMESSAGE+SYS$LIBRARY:LIB.MLB/LIB
c                    $ MAC XALINK
c
c The fortran program must also be compiled:
c
c            $ FOR DR11_snd_rcv
c
c The following link command must be used:
c
c            $ LINK DR11_snd_rcv+XAMESSAGE+XALINK,LINKER/OPT
c
c The program is now ready to run:
c
c            $ RUN DR11_snd_rcv
c
c*************************************************************

 PROGRAM DR11_SND_RCV

c*************************************************************

C#          implicit none

 include '($secdef)'
 include '($ssdef)'

 INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
 INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
 INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
 INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
 INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'

 INCLUDE 'SYS$SHARE:MOVGBL.INC/LIST'   !T.A.B. 4/7/88

 CHARACTER CLUSTER_NAME*8 /'CLUSTER3'/ ! event flag cluster

 CHARACTER*6 HAC1SEC /'HACSEC'/

 INTEGER*4 SYS$MGBLSC, SYS$ASCEFC, SYS$WAITFR, SYS$CLREF
 INTEGER*4 SEC_FLAGS, MAPRANGE(2), RETADR(2)
```

```
 integer*4 indx, status, value, xmit_operation
 integer*4 input_buffer(400), output_buffer(400)
 integer*4 buffer_size, local, remote
 integer*4 i_mox(10),i_moy(10),i_moz(10)

 real*4 r_mou(10),r_mophi(10),r_mohead(10),r_mothet(10)

 logical   test, master

 character*4      c_key_input

c... This common contains the data to be sent to the VAX 11/780

 common /zzl_snd_data/
     &        r_airspeed, r_torque, r_rclimb,
     &        r_altitude, r_roll, r_pitch,
     &        r_yaw, r_slip, r_cosphi,
     &        r_sinphi, r_ugnd, r_vgnd,
     &        r_der, r_dnr, r_ve, r_vn,
     &    r_phi, r_thet, r_psi, r_turn_rate,
     &    i_mox, i_moy, i_moz, r_mou,
     &    r_mohead, r_mophi, r_mothet          !360 bytes

c... This common contains the data to be received from the VAX
C    11/780

 common /zzl_rcv_data/
     &    i_hel_world_x, i_hel_world_y,
     &    c_key_input                     !12 bytes

c... This equivalence associates the output buffer with the
C     snd_data common.

 equivalence (output_buffer(1), r_airspeed)

c... This equivalence associates the output buffer with the
C    rcv_data common.

 equivalence (input_buffer(1), i_hel_world_x)

c*****************************************************************
c... begin executable code
c*****************************************************************

c... Initialize variables

 test   = .TRUE.
 master = .TRUE.  ! MicroVAX will be designated the master

 buffer_size = 200 ! buffer size in words
 local = 2  ! device type at local end: 1=drll-w,2=drvll-w
 remote = 1 ! device type at remote end

c...  Associate the event flag cluster
```

```
 status = sys$ascefc(%val(96), cluster_name,,)
 if (.NOT. status) call lib$stop(%val(status))

c... Set map range for global section

 maprange(1) = %loc(arm(1))
 maprange(2) = %loc(waypt_err)

c... set up flag word

 sec_flags = sec$m_wrt

c... Map to global section

 status = sys$mgblsc (maprange, retadr,, %val(sec_flags),
    &                       HAC1SEC,,)
 if (.NOT. status) call lib$stop(%val(status))

 do while (test)

c...  Wait for the event flag (97).

 status = sys$waitfr (%val(97))
 if (.NOT. status) call lib$stop (%val(status))
 status = sys$clref(%val(97))
 if (.NOT. status) call lib$stop (%val(status))
c...  Update the common area

 CALL UPDATE_TO_VAX

c...  Tranfer direction control: 1 for receive; 0 for transmit
c... 'Transmitting data'

 xmit_operation = 0 ! set up to transmit data

 call DR11_SEND_RECEIVE (
    &                              xmit_operation,
    &                              output_buffer,
    &                              buffer_size,
    &                              local,
    &                              remote,
    &                              master
    &                             )
c... 'Receiving data'
c...  Update the common area

 CALL UPDATE_FROM_VAX

 xmit_operation  = 1 ! set up to receive data

 call DR11_SEND_RECEIVE (
    &                              xmit_operation,
    &                              input_buffer,
```

```
     &                                buffer_size,
     &                                local,
     &                                remote,
     &                                master
     &                    )

      end do

      stop
      end
```

```
C***************************************************************
            Program HACMAIN_BH          !7 december 1987 mcl
C   Author:  NASA Ames equations.
C
C    This program runs flight dynamic equations.
c
c   Subroutines called:
c           Get_chan
c           Hacdata
c           Scale_ad
c           Smart
c           Contrl
c           Fam
c           Beeper
c       Vbturb
c           Wbturb
c           Ubturb
c       sys$creprc, sys$crmpsc, status
c           sys$ascefc, sys$waitfr, sys$clref
c
c   Files opened:
c           Hacsec.dat-----------------'unknown'
c           [helhac.for]Scalars.dat--------'old'
c
c   Files included:
c            [helhac.for]hacl.inc
c            [helhac.for]hac2.inc
c            [helhac.for]hac3.inc
c            [helhac.for]helgbl.inc
c            [helhac.for]fltdata.inc
c
c            $secdef
c            $ssdef
c            $iodef
c
c   Global section name:
c           HACSEC
c
c   Common section name:
c           ACHANNEL
c
c   Cluster name:
c           CLUSTER3________________Flag 96-127
c**************************************************************
  INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
      INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
  INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
  INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
  INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'

  INCLUDE '($SECDEF)'
  INCLUDE '($SSDEF)'
```

```
      INCLUDE '($IODEF)'

      INTEGER*4 SYS$CREPRC,SYS$CRMPSC,STATUS
      INTEGER*4 SEC_FLAGS,MAPRANGE(2),RETADR(2)
      INTEGER*4 GET_CHAN
      INTEGER*2 SEC_CHAN

      CHARACTER*6 HAC1SEC/'HACSEC'/

      EXTERNAL GET_CHAN

      COMMON /ACHANNEL/SEC_CHAN
c...cluster
     INTEGER*4  sys$ascefc
     INTEGER*4  sys$waitfr,sys$clref
     CHARACTER*8      clname3 /'CLUSTER3'/

c   Associate cluster 3
     status = sys$ascefc(%val(96),clname3,,)
     if (.not. status) call lib$stop(%val(status))
C  Create section to share data with DR11 and A/D30Hz processes

      MAPRANGE(1) = %LOC(ARM(1))
      MAPRANGE(2) = %LOC(WAYPT_ERR)

      SEC_FLAGS = SEC$M_GBL .OR. SEC$M_WRT .OR. SEC$M_DZRO

      OPEN(UNIT=1,FILE='HACSEC.DAT',USEROPEN=GET_CHAN,SHARED,
         &        INITIALSIZE=60,STATUS='UNKNOWN')

      STATUS = SYS$CRMPSC(MAPRANGE,RETADR,,%VAL(SEC_FLAGS),
         &   HAC1SEC,,%VAL(SEC_CHAN),%VAL(60),,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

      CALL HACDATA
C  Read A/D scaling factors

     IF(IALLMULAX .EQ. 1)THEN
                 OPEN(UNIT=4,NAME='[HELHAC.FOR]SCALARS_1.DAT',STATUS='OLD')
      READ(4,13)PITSCALAR,ROLSCALAR,PEDSCALAR,COLSCALAR,
         &             PITOFFSET,ROLOFFSET,PEDOFFSET,COLOFFSET

13               FORMAT(8E13.5)
      CLOSE (4)

     ELSE
      OPEN(UNIT=4,NAME='[HELHAC.FOR]SCALARS.DAT',STATUS='OLD')

      READ(4,13)PITSCALAR,ROLSCALAR,PEDSCALAR,COLSCALAR,
         &             PITOFFSET,ROLOFFSET,PEDOFFSET,COLOFFSET
      CLOSE (4)
     ENDIF
     IMODE = -1
         CONTINUE
```

```
 if(iMODE .gt. 0 ) icozero=0

c  Fly
11          CONTINUE
        c - the flag from a/d process

 if (.NOT. status) CALL lib$stop(%VAL(status))
C...Clear the flag
 status = sys$clref(%val(96))
 IF (.NOT. status) CALL lib$stop(%VAL(status))

c...Equations
 call scale_ad
 call smart
 call contrl
 call fam
 call beeper
c                 call dturb

c           call ubturb
 if (abs(FTZ) .GT. wait) imode = 1

 FLIGHTTIME = NINT((SECNDS(0.0))*1000.)

    GOTO 11
 end
```

APPENDIX C

SUBROUTINE SOURCE CODE

XALINK.MAR

```
     .TITLE    XALINK
     .IDENT    /X-1/
;
;*****************************************************************************
;*                                                                          *
;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;*  ALL RIGHTS RESERVED.                                                    *
;*                                                                          *
;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY  *
;*  TRANSFERRED.                                                            *
;*                                                                          *
;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;*  CORPORATION.                                                            *
;*                                                                          *
;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;*                                                                          *
;*                                                                          *
;*****************************************************************************
;
; PROGRAM TO CHANGE DR11W DRIVER FOR LINK MODE
;
;
; SYSTEM MACRO CALLS - DEFINE XADRIVER CHARACTERISTICS BITS
;
     $XADEF
; LOCAL SYMBOLS
     .PSECT    XADATA,LONG
CHANA:     .LONG     0          ; CELL TO STORE CHANNEL NUMBER
IOSB:      .BLKQ     1          ; IOSB FOR QIO
CHAR_BUF:  .LONG     80,INFO        ; DEVICE CHARACTERISTICS BUFFER DESCR
LENGTH:    .LONG     0          ;     AND LENGTH
INFO:      .LONG     0          ; CHARACTERISTICS BUFFER
CHAR:      .LONG     0
DEVDEPEND:      .LONG 0
     .BLKL     20

     .PSECT    XACODE,NOWRT

     .ENTRY    XALINK,^M<R2,R3,R4,R5>

     MOVL @4(AP),W^CHANA        ; GET CHANNEL NUMBER
20$:
     $GETCHN_S CHAN=W^CHANA,-  ; GET CHANNEL INFORMATION
               PRIBUF=W^CHAR_BUF,-
               PRILEN=W^LENGTH
     BLBS R0,40$
```

```
    RET
40$:
    BISL #XA$M_LINK,W^DEVDEPEND    ; SET LINK MODE STATUS BIT
    $QIO_S    CHAN=W^CHANA,-       ; WRITE CHARACTERISTICS
        FUNC=#IO$_SETCHAR,-
        EFN=#10,-
        IOSB=W^IOSB,-
        P1=W^CHAR
    BLBS R0,50$
    RET
50$:
    MOVZWL    W^IOSB,R0
    RET
    .END
```

XAMESSAGE.MAR

```
        .TITLE    XAMESSAGE
        .IDENT    'V04-001'

;****************************************************************************
;*                                                                          *
;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;*  ALL RIGHTS RESERVED.                                                    *
;*                                                                          *
;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;*  TRANSFERRED.                                                            *
;*                                                                          *
;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;*  CORPORATION.                                                            *
;*                                                                          *
;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;*                                                                          *
;*                                                                          *
;****************************************************************************
;
;++
;
; FACILITY:
;
;       VAX/VMS Executive, I/O Drivers
;
; ABSTRACT:
;
;       This module allows you to connect a DR11-W to a DRV11-WA; or
;       a DR11-W to another DR11-W in an interprocessor link and to
;       perform data transfers from one processor to the other.
;
; AUTHOR:  Donald G. Blair
;
; MODIFIED BY:
;
;--
```

```
     .SBTTL    LOCAL DEFFINITIONS AND STORAGE
;++
; XAMESSAGE ROUTINE
;
; CALLING SEQUENCE:
;
;    CALL (BUFFER_ADDRESS,BUFFER_SIZE,TRANSFER_DIRECTION,CHANNEL,-
;          EVENT_FLAG,TIME_OUT,STATUS_ADDRESS,LOCAL_DEVICE,REMOTE_DEVICE)
;
;         BUFFER_ADDRESS = ADDRESS OF DATA BUFFER TO TRANSFER
;         BUFFER_SIZE = SIZE IN BYTES OF DATA BUFFER TO TRANSFER.
;           NOTE THAT RECEIVER AND TRANSMITTER MUST AGREE ON THE
;           SIZE OF THE TRANSFER.
;         TRANSFER_DIRECTION = DIRECTION FOR DATA TO GO
;              0 = TRANSMIT
;              1 = RECEIVE
;         CHANNEL = CHANNEL ASSIGNED TO DEVICE (DR11-W OR DRV11-WA)
;         EVENT_FLAG = EVENT FLAG TO SET WHEN TRANSFER COMPLETE
;         TIME_OUT = I/O TIME-OUT VALUE IN SECONDS
;         STATUS_ADDRESS = ADDRESS OF 20 BYTE ARRAY TO RECEIVE
;           FINAL STATUS - ONLY FILLED IN IF USER'S PARAMETERS ARE
;           ALL VALID.
;              IOSB - 8 BYTES
;                   I/O STATUS BLOCK FROM QUEUE I/O REQUEST
;              ERROR - 4 BYTES - NOT USED - FOR COMPATIBILITY
;                   WITH OLD VERSIONS OF THIS MODULE.
;              STATE - 4 BYTES
;                   THIS FIELD TRACKS WHICH QIO WAS THE LATEST
;                   ONE TO BE PERFORMED.
;                   01 - LAST QIO WAS ONE IN THE MAIN ROUTINE
;                   02 - LAST QIO WAS ONE IN AST_GO.
;              SSRV_STS - 4 BYTES
;                   VALUE OF R0 RETURNED FROM THE LAST SYSTEM
;                   SERVICE EXECUTED.
;         LOCAL_DEVICE = TYPE OF DEVICE AT LOCAL END OF LINK.
;              DR11_W = 1
;              DRV11_WA = 2
;         REMOTE_DEVICE = TYPE OF DEVICE AT REMOTE END OF LINK.
;              DR11_W = 1
;              DRV11_WA = 2
;
;
;--

     $SSDEF

; PARAMETER OFFSETS.

BUFFER_P = 4
BUF_SIZE_P = 8
DIRECTION_P = 12
CHAN_P = 16
EFN_P = 20
TIME_P = 24
STS_ADDR_P = 28
```

```
LCL_DEVICE_P = 32
REM_DEVICE_P = 36

     .PSECT    XADATA,LONG

; SAVED PARAMETER VALUES.

BUFFER:         .LONG     0     ; SAVED BUFFER ADDRESS
BUF_SIZE: .LONG      0     ; SAVED BUFFER SIZE
DIRECTION:      .LONG     0     ; DIRECTION OF TRANSFER
CHAN:           .LONG     0     ; SAVED CHANNEL ASSIGNED TO DR11-W
EFN:      .LONG      0     ; SAVED EVENT FLAG NUMBER
TIME:           .LONG     0     ; SAVED TIME-OUT VALUE
STS_ADDR: .LONG      0     ; ADDRESS OF CALLERS STATUS VARIABLE

; DEFINE DEVICE TYPES AT BOTH ENDS OF INTERPROCESSOR LINK.

DR11_W = 1
DRV11_WA = 2
LCL_DEVICE:     .BLKL     1     ; TYPE OF DEVICE ON THIS SYSTEM.
REM_DEVICE:     .BLKL     1     ; TYPE OF DEVICE AT OTHER END OF LINK.
AST: .BLKL      1


; NOTE - ORDER IS ASSUMED FOR NEXT FOUR VARIABLES

IOSB:     .QUAD      0          ; QIO IOSB
ERROR:    .LONG      0          ; ERROR VALUE PARAMATER
STATE:    .LONG      0          ; STATE VARIABLE
SSRV_STS: .LONG      0          ; SYSTEM SERVICE STATUS

     .PAGE
     .SBTTL    VALIDATE AND SAVE CALLER'S PARAMETERS
     .PSECT    XACODE,NOWRT
     .ENTRY    XAMESSAGE,^M<R2,R3,R4,R5>

; VALIDATE AND SAVE CALLER'S PARAMATERS

     CLRQ W^IOSB          ; CLEAR IOSB
     CLRL W^ERROR         ; CLEAR ERROR FIELD
     CLRL W^SSRV_STS      ; CLEAR SYS SERVICE RETURN STATUS.
     CMPW (AP),#9         ; MUST HAVE 9 PARAMATERS
     BEQL      10$        ; BR IF OKAY
     BRW  BADPARAM        ; BR TO SIGNAL ERROR
10$: MOVL BUFFER_P(AP),W^BUFFER    ; GET BUFFER ADDRESS
     MOVL @BUF_SIZE_P(AP),W^BUF_SIZE ; GET BUFFER SIZE
     BNEQ 20$             ; BR IF OKAY
     BRW  BADPARAM        ; XFER SIZE IS NON ZERO -- ILLEGAL
20$: MOVZBL    @DIRECTION_P(AP),W^DIRECTION ; GET TRANSFER DIRECTION FLAG
     CMPL W^DIRECTION,#2       ; THE ONLY LEGAL VALUES ARE 0,1
     BLEQU     25$             ; BR IF OKAY
     BRW  BADPARAM        ; ELSE BR TO SIGNAL ERROR
25$: MOVL @CHAN_P(AP),W^CHAN  ; FETCH CHANNEL
     MOVL @EFN_P(AP),W^EFN     ; AND EVENT FLAG
     BEQL BADPARAM        ; MUST SPECIFY EVENT FLAG
```

```
        MOVL @TIME_P(AP),W^TIME  ; FETCH TIME-OUT VALUE
        BNEQ 30$              ; IF NONZERO, USE IT.
        MOVZBL    #5,W^TIME       ; ELSE USE SOME "REASONABLE" VALUE
30$:    MOVL STS_ADDR_P(AP),W^STS_ADDR ; GET ADDRESS OF STATUS
                                        ; ARRAY
        BEQL BADPARAM         ; IF NOT SPECIFIED, ERROR
        CLRL @W^STS_ADDR          ; INITIALIZE STATUS VALUE
        MOVZBL    @LCL_DEVICE_P(AP),W^LCL_DEVICE ; GET LOCAL DEVICE TYPE
        CMPL #DRV11_WA,W^LCL_DEVICE   ; IS LOCAL DEVICE A DRV11-WA?
        BEQLU     35$             ; BRANCH IF SO.
        CMPL #DR11_W,W^LCL_DEVICE     ; IS LOCAL DEVICE A DR11-W?
        BNEQU     BADPARAM        ; ERROR IF IT'S NOT EITHER.
35$:    MOVZBL    @REM_DEVICE_P(AP),W^REM_DEVICE ; GET REMOTE DEVICE
                                            ;TYPE
        CMPL #DRV11_WA,W^REM_DEVICE   ; IS REMOTE DEVICE A DRV11-WA?
        BEQLU     50$             ; BRANCH IF SO.
        CMPL #DR11_W,W^REM_DEVICE     ; IS REMOTE DEVICE A DR11-W?
        BNEQU     BADPARAM        ; ERROR IF IT'S NOT EITHER.
50$:    $CLREF_S EFN=EFN          ; MAKE SURE EFN IS CLEAR
        BLBS R0,100$          ; BR IF NO SYS SERVICE ERROR
        RET
100$:        CMPL #DRV11_WA,W^LCL_DEVICE   ; DISPATCH BASED ON LOCAL DEVICE TYPE
        BEQL      DRV11_WA_START      ; LOCAL DEVICE IS DRV11-WA
        BRW  DR11_W_START         ; LOCAL DEVICE IS DR11-W

BADPARAM:
        MOVZWL    #SS$_BADPARAM,R0    ; ELSE RETURN ERROR.
        RET


        .PAGE
        .SBTTL    START MESSAGE PROCESSOR

DRV11_WA_START:                       ; THE LOCAL DEVICE IS A DRV11-WA
        BLBC W^DIRECTION,10$          ; BRANCH IF IT'S A XMIT OPERATION
        MOVAL     W^AST_COMPLETION,W^AST    ; AST_COMPLETION IS THE AST FOR RECEIVE
        BRB  20$
10$:    MOVAL     W^AST_GO,W^AST      ; AST_GO IS THE AST FOR XMIT OPERATION
20$:    MOVL #01,W^STATE          ; STATE = 1 => LAST QIO WAS IN MAIN
                              ;    ROUTINE.
        $QIO_S    CHAN=W^CHAN,-       ; BLOCK MODE READ - EVEN IF IT'S XMIT
             FUNC=#<IO$_READLBLK!IO$M_TIMED!IO$M_SETFNCT>,-
             IOSB=W^IOSB,-
             ASTADR=@W^AST,-
             P1=@W^BUFFER,-           ; ADDRESS OF CALLER'S DATA BUFFER
             P2=W^BUF_SIZE,-      ; LENGTH OF DATA BUFFER
             P3=W^TIME,-          ; TIMEOUT VALUE
             P4=#7                ; INTERRUPT+READ
        BRW  MAIN_EXIT        ; EXIT MAIN ROUTINE.

DR11_W_START:                     ; LOCAL DEVICE IS DR11-W
        MOVL #01,W^STATE          ; STATE = 1 => LAST QIO WAS IN MAIN
                              ;    ROUTINE.
        $QIO_S    CHAN=W^CHAN,-       ; QIO TO ENABLE AST'S
             FUNC=#<IO$_SETMODE!IO$M_ATTNAST>,-
```

```
          IOSB=W^IOSB,-
          P1=W^AST_GO
     BLBC RO,MAIN_EXIT          ; BRANCH ON ERROR - ALL DONE.
     BLBS W^DIRECTION,MAIN_EXIT    ; BRANCH IF IS A RECEIVE OPERATION
     CMPL #DR11_W,W^REM_DEVICE     ; IS REMOTE DEVICE A DR11-W?
     BNEQU     MAIN_EXIT           ; BRANCH IF NOT.
     $QIO_S    CHAN=W^CHAN,-       ; PERFORM 0-LENGTH QIO.  THIS
          FUNC=#<IO$_WRITELBLK!IO$M_SETFNCT>,- ; SERVES TO SET THE
          IOSB=W^IOSB,-         ; FNCT BITS (CONTAINED IN P4),
          P1=@W^BUFFER,-        ; IN THE CSR, INTERRUPTING THE REMOTE
          P2=#0,-               ; DR11-W.
          P4=#2

MAIN_EXIT:
     MOVL RO,W^SSRV_STS         ; SAVE QIO STATUS RETURN
     MOVC3     #20,W^IOSB,@W^STS_ADDR   ; RETURN STATUS TO THE USER
     BLBS W^SSRV_STS,10$        ; IF SUCCESS, DON'T SET EVFLAG YET
     $SETEF_S EFN=W^EFN         ; IF ERROR, SET EVENT FLAG -- ALL DONE.
10$: MOVL W^SSRV_STS,RO         ; RESTORE RO STATUS RETURN.
     RET


     .PAGE
     .SBTTL    AST_GO - AST WHICH INITIATES THE QIO TO PERFORM ACTUAL XFER.
     .ENTRY    AST_GO,^M<R2,R3,R4,R5>

;
; This AST is called to perform the $QIO which begins the actual transfer
; of user data.  (Hence the name AST_GO.)
;
     BLBS W^DIRECTION,AST_RECEIVE ; BRANCH IF RECEIVE OPERATION


;
; On a DR11-W, this AST is delivered as a result of an interrupt from the
; remote device, so no status checking is necessary.  On a DRV11-WA, this
; AST is delivered as a result of an intentionally premature I/O
; completion, so we expect the status return to be SS$_OPINCOMPL.
;
AST_XMIT:
     CMPL #DRV11_WA,W^LCL_DEVICE    ; IS LOCAL DEVICE A DRV11-WA?
     BNEQ 20$                  ; BRANCH IF NOT.
     CMPW W^IOSB,#SS$_OPINCOMPL    ; STATUS SHOULD BE SS$_OPINCOMPL.
     BEQL 20$                  ; BR IF EXPECTED STATUS
     BRW  IO_DONE                 ; ELSE ERROR
20$: MOVL #02,W^STATE          ; STATE = 2 => LAST QIO WAS IN
                               ; AST_GO.
     $QIO_S    CHAN=W^CHAN,-       ; BLOCK MODE WRITE
          FUNC=#<IO$_WRITELBLK!IO$M_TIMED!IO$M_SETFNCT!IO$M_CYCLE>,-
          IOSB=W^IOSB,-
          ASTADR=W^AST_COMPLETION,-
          P1=@W^BUFFER,-        ; ADDRESS OF CALLER'S DATA BUFFER
          P2=W^BUF_SIZE,-          ; LENGTH OF BUFFER
          P3=W^TIME,-           ; TIMEOUT VALUE
          P4=#4                 ; FNCT BITS FOR CSR
     BLBS RO,40$                ; RETURN IF QIO STARTED OK
```

```
	BRW  IO_DONE                ; ALL DONE IF ERROR OCCURRED.
40$: RET              ; DISMISS THIS AST, AND
                           ; WAIT FOR AST_COMPLETION
;
; AST_RECEIVE is only used by the DR11-W, since the DRV11-WA initiates
; the actual data transfer from the main routine when it is the receiver.
;
AST_RECEIVE:
	MOVL #02,W^STATE     ; STATE = 2 => LAST QIO WAS IN AST_GO.
	$QIO_S    CHAN=W^CHAN,-        ; BLOCK MODE READ
		FUNC=#<IO$_READLBLK!IO$M_TIMED!IO$M_SETFNCT>,-
		IOSB=W^IOSB,-
		ASTADR=W^AST_COMPLETION,- ; ADDRESS OF AST FOR I/O
		                  ;COMPLETION
		P1=@W^BUFFER,-            ; ADDRESS OF CALLER'S DATA BUFFER
		P2=W^BUF_SIZE,-      ; LENGTH OF DATA BUFFER
		P3=W^TIME,-          ; TIMEOUT VALUE
		P4=#7                ; INTERRUPT+READ
	BLBS R0,10$              ; RETURN IF QIO STARTED OK
	BRW  IO_DONE             ; ON ERROR, WE'RE ALL DONE.
10$: RET


	.PAGE
	.SBTTL    AST_COMPLETION - COMPLETION ROUTINE FOR I/O TRANSFER.
	.ENTRY AST_COMPLETION,^M<R2,R3,R4,R5>

;
; This AST is called when the actual transfer of data is complete.  Note
; that the status value in the IOSB must be checked by the caller when
; we're done.
; IO_DONE is also called when an error occurs and the handshaking
; sequence must be terminated.
;

IO_DONE:
	MOVC3     #20,W^IOSB,@W^STS_ADDR    ; RETURN STATUS TO THE USER
	$SETEF_S EFN=W^EFN          ; SET THE CALLER'S EVENT FLAG
	MOVZBL    #SS$_NORMAL,R0        ; SIGNAL SUCCESSFUL AST
		              ;COMPLETION.
	RET
	.END
```

```
c*****************************************************************

      SUBROUTINE DR11_SEND_RECEIVE (
     &                         xmit_operation,
     &                         data_buffer,
     &                         buffer_size,
     &                         local,
     &                         remote,
     &                         master
     &                         )

c*****************************************************************

      implicit none

c*****************************************************************
c               VARIABLE DECLARATIONS
c*****************************************************************

      integer*4  local,remote,buffer_size
      integer*4  status,nchan,indx
      integer*4  time /30/
      integer*4  xmit_operation
      integer*4  iosb(10)
      integer*4  sys$assign,sys$waitfr
      integer*4  xamessage,xalink
      integer*4  data_buffer(400)

      logical    linked /.FALSE./
      logical    master

c*****************************************************************
C               CHARACTER DECLARATIONS
c*****************************************************************

      character  DEVICE*5/'XAA0:'/

      external  xalink,xamessage

c*****************************************************************
c               Begin Executable Code
c*****************************************************************

c... assign channel to DR11-W and place xadriver in LINK mode for
C    this channel

      if (.NOT. linked) then
        status=sys$assign('XAA0:',nchan,,)
        if(.not. status)goto 100
        status=xalink(nchan)
        if(.not. status)goto 150
        linked = .TRUE.
```

```
      endif

10    continue

c call xamessage routine to exchange data;
c... xmit_operation = 1 for receive, 0 for transmit

         status=xamessage(
      &         data_buffer,
      &         buffer_size*2,
      &         xmit_operation,
      &         nchan,
      &         12,
      &         time,
      &         iosb,
      &         local,
      &         remote )
      if(.not. status)goto 200

      status=sys$waitfr(%val(12))
      if(.not. status)goto 300

      goto 50 ! let's exit go back for next iteration

c... Error messages

100   write(6,1010)status
1010  format(' error from assign  ',i8)
      call exit
150   write(6,1015)status
1015  format(' error from xalink  ',i8)
      call exit
200   write(6,1020)status
1020  format(' error from xamessage  ',i8)
      goto 50
300   write(6,1030)status
1030  format(' error from waitfr  ',i8)

50    continue

      return
      end
```

UPDATE_TO_VAX.FOR

```
c***************************************************************

      SUBROUTINE UPDATE_TO_VAX

c***************************************************************

      INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
      INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
      INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
      INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
      INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'

      INCLUDE 'SYS$SHARE:MOVGBL.INC/LIST'     !T.A.B. 4/7/88

      integer*4 i_mox(10),i_moy(10),i_moz(10)

      real*4 r_mou(10),r_mophi(10),r_mohead(10),r_mothet(10)


c... This common contains the data to be sent to the VAX 11/780

      common /zzl_snd_data/
     &      r_airspeed, r_torque, r_rclimb,
     &      r_altitude, r_roll, r_pitch,
     &      r_yaw, r_slip, r_cosphi,
     &      r_sinphi, r_ugnd, r_vgnd,
     &      r_der, r_dnr, r_ve, r_vn,
     &      r_phi, r_thet, r_psi, r_turn_rate,
     &      i_mox, i_moy, i_moz, r_mou,
     &      r_mohead, r_mophi, r_mothet

C... Begin executable code.

      r_airspeed          = UB
      r_torque            = QRAPCT
      r_rclimb            = ALTD
      r_altitude          = ALT
      r_roll              = PHIR
      r_pitch             = THETR
      r_yaw               = PSIR
      r_slip              = VB
      r_cosphi            = CPHI
      r_sinphi            = SPHI
      r_ugnd              = UB
      r_vgnd              = VB
      r_der               = DER
      r_dnr               = DNR
      r_ve                = VE
      r_vn                = VN
      r_phi               = PHI
      r_thet              = THET
      r_psi               = PSI
```

```
      r_turn_rate     = TURN_RATE    !MCL 07-22-1988 GLH

      DO 45 I=1,10
         i_mox(i)     = MOX(I)
         i_moy(i)     = MOY(I)
         i_moz(i)     = MOZ(I)
         r_mou(i)     = MOU(I)
         r_mohead(i) = MOHEAD(I)
         r_mophi(i)  = MOPHI(I)
         r_mothet(i) = MOTHET(I)
45       CONTINUE

      return
      end
```

UPDATE_FROM_VAX.FOR

```
c****************************************************************

      SUBROUTINE UPDATE_FROM_VAX

c****************************************************************

      INCLUDE '[HELHAC.FOR]HELGBL.INC/LIST'
      INCLUDE '[HELHAC.FOR]FLTDATA.INC'
c...  This common contains the data to be received to the VAX
C     11/780
      character*4     c_key_INPUT

      common /zzl_rcv_data/
      &    i_hel_world_x, i_hel_world_y,
      &    c_key_input

C...  Begin executable code.

      hel_world_x         = i_hel_world_x
      hel_world_y         = i_hel_world_y
      key_input           = c_key_input

      return
      end
```

HACDATA.FOR

```
                SUBROUTINE HACDATA
C*************************************************************
C               HAC DATA INITIALIZATION
C*************************************************************

      INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
      INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
      INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
      INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'


        G = 32.174                      !adDED 6/23/87
        HLEV=36089.
        XM1=1.
        PRESZ=2116.2
        TEMPZ=518.69
c        YCON=100.0                      !changed due to problems
                                         !with rho MCL 19-30-1987
        YCON = 1.0
        TEMA=.59249
        ICOND=1                         ! ADDED 6/16/87 GLH
        RHO=.0023671                    !slug/ft3 added 6/16/87
        RHOZ=.0023671                   !SLUG/FT3 ADDED 6/23/87 GLH
        SOUND=1140.                     ! ADDED 6/16/87 GLH
        SOUNDZ=1140.                    !ADDED 6/16/87 GLH
        IETURB=0                        !ADDED 6/17/87 GLH. Changed
c                                        to 0
c                                        3/31/88 GLH.
        XP=4.58330                      !ADDED 6/17/GLH
        YP=0.0                          !ADDED 6/17/87 GLH
        ZP=0.0                          !ADDED 6/17/87 GLH
        XMASS=497.36                    !ADDED 6/19/87 GLH
        WAIT=16002.                     !ADDED 6/24/87 GLH
        RE=20924640.                    !radius of earth in feet
C                                        added 6/23/87 GLH
        RR = RE                         !RAD. OF EARTH + HT. OF
C                                        RUNWAY 8/26/87 TAF
        IFLAT=1                         !ADDED 6/23/87 GLH
        DT2=0.06                        !ADDED 6/19/87 GLH,CHANGED FROM .03
TO
C                                        .06 7/21/88
        CTHT=1.0                        !ADDED 6/23/87 GLH
        R2D=57.2957795                  !ADDED 6/23/87 GLH

C-----PILOT INPUTS,LIMITS, DEADBANDS, GAINS
C
            ROLLO     =     0.0
            PITCHO    =     0.0
            YAWO      =     0.0
            ROLMAX    =    50.
            PITMAX    =    50.
```

```
          YAWMAX     =    50.
          COLOMIN    =  -100.0
          COLOMAX    =   100.0
          RDBO       =     6.0      !CHANGED 9/16/87 GLH
          PDBO       =     6.0      !CHANGED 9/16/87 GLH
          YDBO       =     3.0      !CHANGED 9/16/87 GLH
          CDBO       =     0.0
C-----GAINS FOR +-50( OOO = 50/INCHES OF TRAVEL)
C          OOOG1      =     8.33
          OOOG1      =    5.33           !roll
C          OOOG2      =     8.33
          OOOG2      =    3.00           !pitch
C          OOOG3      =    15.625
          OOOG3      =   3.5
C          OOOG4      =   -10.0
          OOOG4      =   -9.0   ! GLH 5 OCT 87
C          OOOG4      =   -12.0      !mcl 19-aug-1987
          OOOB4      =     0.0
C-----MAG BRAKE AND BEEPER TRIM
          IMBC       =    0
          ILITE      =    1
          GNUD       =    0.5
          GRLD       =    0.5
C-----LOADER RATE LIMIT ( INCHES/CYCLE )
      TINCIC          =  1.0
      TINCOP          =  0.25
C-----INITIAL TRIM POINT
      TRIM1  = 0.
      TRIM1C = 0.
      TRIM1P = 0.
      TRIM2  = 0.
      TRIM2C = 0.
      TRIM2P = 0.
      TRIM3  = 0.
      TRIM3C = 0.
      TRIM3P = 0.
      TRIM4  = 0.
      TRIM4C = 0.
      TRIM4P = 0.
C-----ICOZERO=1 FOR SUBTRACTING IC VALUES FROM PILOT INPUTS
      ICOZERO     =     1
C-----COLLECTIVE POSITION LIMITS  AND %BIAS.
      COLMINT = 0.0
      COLMAXT = 10.0
      PCTBIAS = 0.0
C...GRADIENTS, BREAKOUTS, HYSTERISIS---NOMINAL VALUES.
      PEDBRK = 2.0
      PEDGRD = 2.0
      PCHBRK = 1.0
      PCHGRD = 0.67
      ROLBRK = 1.0
      ROLGRD = 1.0
      CLCFRC = 0.5
      CLCGRD = 0.0
      CLCBRK = 0.0
```

```
       CBAUTO = 0.0
       CGAUTO = 0.0
       CFMAN  = 10.0
       CFAUTO = 1.0
       COLHYS = 0.0
       ROLHYS = 0.0
       PEDHYS = 0.0
C-----CONTROL SYSTEM OPTION FLAG.
       ICS   = 2
       MCS   = 1
       ICONFI  =  2
       IALLMULAX = 0 ! when set to 1 multiaxis
C                      roll,pitch, yaw,and collective are used.
       XNV    =  2.50       !CHANGED 6/17/87 GLH
       XRT    =  0.0
       YRT    =  0.0
       XUT    = -0.01
       YVT    = -0.1
       ZWT    = -1.0
       ZDCT   =  1.5
       ZLPT   = -2.8
       ZMQT   = -2.8
       ZNRT   = -2.0
       XNDPT =  0.04
       XKDP   =  0.0209
       FPS2KOS =  0.592485
       ZUO    =  1.0
       XLPHIO = -6.25
C-----SENSITIVITY CONSTANTS.
       ZLVT = 1.0
       ZMUT = 0.3  ! Changed from 1.0 to 0.3 4/4/88 GLH
       ZMWT = 0.2  ! Changed from 1.0 to 0.2 4/4/88 GLH
       ZNVT = 1.0
       ZUT  = 1.0
       ZWTT = 1.0
       ZMU  = 0.0
       ZLV  = 0.0
C-----CONTROL SYSTEM DEPNDENT PARAMETERS
C-----ICS =      1,      2,      3
       XLPHI1 = 0.0
       XLPHI2 = -6.25
       XLPHI3 = -6.25
       ZLPD    = -5.6
       ZMQD    = -5.6
       ZMQ1    = -5.6
       ZMQ2    = -3.5
       ZLP1    = -5.6
       ZLP2    = -3.5
       XLDA1   = 0.2
       XLDA3   = 0.2
       XMDE1   = 0.14
       XMDE3   = 0.1
C-----FTZ LIMITING
C-----V-N CURVE  FOR UH-60 HELICOPTER.
       VQMAX(1) = 0.0
```

```
      RTMAX(1) = 3.3
      VQMAX(2) = 120.0
      RTMAX(2) = 2.42
      VQMAX(3) = 160.0
      RTMAX(3) = 2.10
      VQMAX(4) = 180.0
      RTMAX(4) = 1.58
      VQMAX(5) = 190.0
      RTMAX(5) = 1.58
      VQMAX(6) = 200.0
      RTMAX(6) = 1.58
      VQMIN(1) = 0.0
      RTMIN(1) = 0.6
      VQMIN(2) = 54.0
      RTMIN(2) = 0.0
      VQMIN(3) = 154.0
      RTMIN(3) = 0.0
      VQMIN(4) = 180.0
      RTMIN(4) = 0.5
      VQMIN(5) = 190.0
      RTMIN(5) = 0.5
      VQMIN(6) = 200.0
      RTMIN(6) = 0.5
C-----COLLECTIVE TRIM . ( 13MAY85).
      SLP1  = 0.25
      YINT1 = 50.6
      SLP2  = -0.25
      YINT2 = 50.6
      SLP3  = 0.0
      YINT3 = 40.8
      SLP4  = 0.27
      YINT4 = 16.5
      SLP5  = 0.72
      YINT5 = -46.7
C-----AIRCRAFT CONFIGURATION:
C-----ICON=1         .LOW BASELINE.    AH-1S.
C          2         .HIGH BASELINE.   UH-60.
C          3         .LHX.             LHX.
C-----RATE OF CLIMB .
      RSLP1(1) = 0.176
      RSLP1(2) = 0.11
      RSLP1(3) = 0.08
      XINT1(1) = 14.4
      XINT1(2) = 31.1
      XINT1(3) = 38.8
      RSLP2(1) = -0.192
      RSLP2(2) = -0.252
c       RSLP2(2) = -0.150      !mcl 19-aug-1987
      RSLP2(3) = -0.252
      XINT2(1) = 40.13
      XINT2(2) = 56.42
c       XINT2(2) = 70.00       !mcl 19-aug-1987
      XINT2(3) = 62.03
C-----SIDESLIP LIMITATION ENVELOPE .
      ESLP1(1) = -2.0
```

```
      ESLP1(2) = -1.082
      ESLP1(3) = -0.932
      EINT1(1) = 150.0
      EINT1(2) = 138.68
      EINT1(3) = 145.92
      ESLP2(1) = -0.16
      ESLP2(2) = -0.146
      ESLP2(3) = -0.146
      EINT2(1) = 30.4
      EINT2(2) = 40.4
      EINT2(3) = 51.6
      VEQPT(1) = 65.0
      VEQPT(2) = 105.0
      VEQPT(3) = 120.0
C...MAXIMUM RATE OF DESCENT .
      RODSLP  = 0.114286
      RODINT1 = -20.0
      RODINT2 = -4.00
C-----INITIAL THETA FOR ZRT CALCULATION.
      THETIC  = 0.0
      THETVIC = 0.0
C...THETA OFFSET FOR INITIALIZATION PURPOSES.
      THETOFF = 0.0
C-----COLLECTIVE/ENGINE FILTER TIME CONSTANT AND GAIN
      TAUCOL = 0.1
      GCOL   = 1.0
C-----PERCENT LIMIT FOR FTZ APPROACH.
C-----TIME DELAY FOR ONSET OF FTZ RED G-LIMIT LIGHT:
      APPPCT   =  10.0
      REDTIME  =   0.5
C-----SOUND GENERATOR VOLUMES
      SGVOLIC = 0.0
      SGVOLOP = 1.5
C...I.C. COLLECTIVE TRIM SWITCH.
      LTRIM    =   0
C-----BETAHUD FUDGE FACTOR
      BETAFG  =     .4678
      D2R     =     .01745     !ADDED 6/23/87

     XLONR = 350. * D2R   !ADDED 8/26/87
     XLATR =  50. * D2R   !ADDED 8/26/87
     CLATR = COS(XLATR)
C
C-----HEADING INITIALIZATION
C
     PSIR = 270.0/R2D    !.............Added MCL 01/22/1988
C
     RETURN
     END
```

SCALEAD.FOR

```
C***************************************************************
      SUBROUTINE SCALEAD
c    Author:  Finley
c
c    This program scales the input from the A/D
C
c
c    Subroutines called:
c     none
c
c    Files opened:
c     none
c
c    Files included:
c     [HELHAC.FOR]HAC1.INC
c     [HELHAC.FOR]HAC2.INC
c     [HELHAC.FOR]HAC3.INC
c     [HELHAC.for]HELGBL.INC
c     [HELHAC.FOR]FLTDATA.INC
c
c    Global section name:
c     HACSEC
c
c     INPUTS:
c      chad(1)________________Pitch
c      chad(2)________________Roll
c      chad(3)________________Collective (conventional)
c      chad(4)________________Pedals (conventional)
c      chad(5)----------------Collective from multiaxis control
c      chad(6)----------------Yaw from multiaxis
c      pitscalar,rolscalar,
c      colscalar,pedscalar
c      colscalar_mulax,
c      pedscalar_mulax____Scaling factors.
c      pitoffset,roloffset,
c      coloffset,pedoffset
c      coloffset_mulax,
c      pedoffset_mulax____Offset.
c      imode__________________Flight mode.
c
c     OUTPUTS:
c      pitchop________________Pitch.
c      rollop_________________Roll.
c      colop__________________Collective.
c      yawop__________________Pedals.
c***************************************************************
      INTEGER*4 torquep
      INCLUDE   '[HELHAC.FOR]HAC1.INC/LIST'
      INCLUDE   '[HELHAC.FOR]HAC2.INC/LIST'
      INCLUDE   '[HELHAC.FOR]HAC3.INC/LIST'
      INCLUDE   '[HELHAC.for]HELGBL.INC/LIST'
      INCLUDE   '[HELHAC.FOR]FLTDATA.INC/LIST'
```

```
      IF ((IMODE .EQ. -1) .OR.
     &      (IMODE .EQ.  1)) THEN
C...Added by MCL 01/26/1988
      IF (ALT .LE. 0.0) THEN
       IF (UB .GT. 1.0) THEN
        CHAD(1) = 800
       ELSE
        IF (UB .LT. -1.0) THEN
         CHAD(1) = -800
        ELSE
         CHAD(1) = 0
        ENDIF
       ENDIF
       CHAD(2) = 0
       CHAD(4) = 0
       CHAD(6) = 0
      ENDIF
C...
     CALL GET_AD_VAL(torquep)
c...
        PITCHOP =PITSCALAR * CHAD(1) + PITOFFSET
        ROLLOP = ROLSCALAR * CHAD(2) + ROLOFFSET
c...Added by GLH 6/22/88 to use full multiaxis inputs.
c...If IALLMULAX is 1 then the conventional pedals and
c...collective
c...controls are not used. All multiaxis control features
c...are used.
        IF(IALLMULAX .EQ.0)THEN
             COLOP = COLSCALAR * CHAD(3) + COLOFFSET
             YAWOP = PEDSCALAR * CHAD(4) + PEDOFFSET
        ELSE
          COLOP = COLSCALAR * CHAD(5) + COLOFFSET
          YAWOP = PEDSCALAR * CHAD(6) + PEDOFFSET
        ENDIF
         IF(PITCHOP.GT.10.)PITCHOP=10.
         IF(PITCHOP.LT.-10.)PITCHOP=-10.
         IF(ROLLOP.GT.10.)ROLLOP=10.
         IF(ROLLOP.LT.-10.)ROLLOP=-10.
         IF(COLOP.GT.10.)COLOP=10.
         IF(COLOP.LT.0.)COLOP=0.
         IF(YAWOP.GT.10.)YAWOP=10.
         IF(YAWOP.LT.-10.)YAWOP=-10.
      ELSE             !When trimming
         PSTK =  PITSCALAR * CHAD(1) + PITOFFSET
         RLSTK = ROLSCALAR * CHAD(2) + ROLOFFSET
         IF(IALLMULAX .EQ.0)THEN
          CLSTK = COLSCALAR * CHAD(3) + COLOFFSET
          RDSTK = PEDSCALAR * CHAD(4) + PEDOFFSET
         ELSE
          CLSTK = COLSCALAR * CHAD(5) + COLOFFSET
          RDSTK = PEDSCALAR * CHAD(6) + PEDOFFSET
         ENDIF
C
C
```

```
      IF(PSTK.GT.10.)PSTK=10.
      IF(PSTK.LT.-10.)PSTK=-10.
      IF(RLSTK.GT.10.)RLSTK=10.
      IF(RLSTK.LT.-10.)RLSTK=-10.
      IF(IALLMULAX .EQ. 0)THEN
          IF(CLSTK.GT.10.)CLSTK=10.
          IF(CLSTK.LT.0.)CLSTK=0.
      ELSE
       IF(CLSTK.GT.10.)CLSTK=10.
       IF(CLSTK.LT.-10.)CLSTK=-10.
      ENDIF
      IF(RDSTK.GT.10.)RDSTK=10.
      IF(RDSTK.LT.-10.)RDSTK=-10.
   ENDIF
   RETURN
   END
```

GET_AD_VAL.FOR

```
C***************************************************************
      Subroutine get_ad_val(torquep)
C***************************************************************

      IMPLICIT NONE

      INTEGER*4 torque_max_offset
      INTEGER*4 torque_min_offset
      INTEGER*4 torque_change
      INTEGER*4 torquep
      INCLUDE        '[helhac.for]hac3.inc'
c.............................................................
      torque_max_offset = 100
      torque_min_offset = -100

c...Find the change in torque
      IF (chad(5) .GT. torque_max_offset) THEN
       torque_change = chad(5) - torque_max_offset
      ELSE
       IF (chad(5) .LT. torque_min_offset) THEN
        torque_change = chad(5) - torque_min_offset
       ELSE
        torque_change = 0
       ENDIF
      ENDIF
c...Apply the change in torque
      chad(5) = torquep + torque_change
      IF (chad(5) .GT. 3000) chad(5) = 3000
      IF (chad(5) .LT. -3000) chad(5) = -3000
      torquep = chad(5)

      RETURN
      END
```

CONTRL.FOR

```
C     TITLE                                                      CONTRL
C
C
      SUBROUTINE CONTRL
C
C
C
C***********************************************************
C************    SUBROUTINE ABSTRACT AND HIERARCHY   *******
C***********************************************************
C
C
C   ROUTINE PROVIDES FOR HELICOPTER CONTROLS.
C   IT REQUIRES A MODIFIED SMART.
C
C
C***********************************************************
C************       CREATION AND MODIFICATION LOG    *******
C***********************************************************
C
C
C  6/25/87 CREATED FROM CONTR2.FOR ,GLH
C***********************************************************
C*************        SIGNIFICANT VARIABLES     ************
C
C
C----------------
C-- INPUTS
C----------------
C
C  ROLLOP     :LATERAL CYCLIC        +- 6   INCHES
C  PITCHOP    :LONGITUDINAL CYCLIC   +- 6   INCHES
C  YAWOP      :PEDAL                 +- 3.5 INCHES
C  COLOP      :COLLECTIVE            0 - 10 INCHES
C
C
C ICS         CONTROL SYSTEM TYPE:  RATE COM/ATT. HOLD
C                                   PURE RATE COMMAND
C                                   ATT. COM./ATT. HOLD.
C ICONFI    CRAFT CONFIGURATION:  UH60, AH1S OR LHX.
C
C
C APPPCT      APPROACH PERCENT, 0-100
C             PERCENT OF FULL ENGINE POWER FOR YELLOW LIGHT
C REDTIME     TIME AT 100 PERCENT TORQUE FOR RED LIGHT
C
C
C
C
          INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
                INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
```

```
C
C**********************************************************
C*****************   EXECUTABLE  CODE   ******************
C**********************************************************
C
C
      IF ( IMODE ) 10 , 9999 , 100
C**********************************************************
C***************   I.  C.   SECTION   ********************
C**********************************************************
C
  10  CONTINUE
C
C
C-----CALCULATE SLOPES AND INTERCEPTS FOR FTZ LIMITS.
C
      DO 15 JJ=1,5
         SLPMX(JJ) =(RTMAX(JJ+1)-RTMAX(JJ))/(VQMAX(JJ+1)-VQ
         SLPMN(JJ) =(RTMIN(JJ+1)-RTMIN(JJ))/(VQMIN(JJ+1)-VQ
         RYINT(JJ) = RTMAX(JJ) - SLPMX(JJ)*VQMAX(JJ)
         YMINT(JJ) = RTMIN(JJ) - SLPMN(JJ)*VQMIN(JJ)
  15  CONTINUE
C
C
C-----FTZ % APPROACH FACTOR.
C
      FAPPCT=APPPCT*.01
C
C
C-----INITIALIZE CONSTANT FOR USE IN FTX,FTZ CALCS.
C-----THESE ARE SMALL PERTERBATION LIMITING, MAYBE
C
      ZRT= -ZWT*WB - G*COS(THETR)
      XRT = G*SIN (THETVIC*D2R)
C
C
C-----PEDAL TO YAW ACCEL SENSITIVITY FACTOR
C-----SIDESLIP LIMITATION ENVELOPE FACTOR
C
      XNDPT    = XKDP*(YVT*ZNRT + XNV )
      SLPFACT = -1.0/XKDP
C
C
C---- ICS = 3 FILTER COEFFICIENTS
C
C-----PITCH AXIS
      AAP(1)=0.0
      AAP(2)=-ZMQD
      BBP(1)=-XMTHETO
      BBP(2)=-ZMQT
      BBP(3)=1.0
C      CALL FACT(2,DT2,AAP,BUFFP)!COMMENTED OUT 6/19/87 GLH
C-----ROLL AXIS
      AAR(1)=0.0
      AAR(2)=-ZLPD
```

```
      BBR(1)=-XLPHIO
      BBR(2)=-ZLPT
      BBR(3)=1.0
C      CALL FACT(2,DT2,AAR,BUFFR)!COMMENTED OUT 6/19/87 GLH
C
C
C-----COLLECTIVE (ENGINE) FILTER COEFFICIENTS.
C-----AND INITIAL FILTER OUTPUT
C
      COLF1=EXP(-DT2/TAUCOL)
      COLF2=GCOL*(1.0-COLF1)
      COLOZ=GCOL*COLOP
C
C
C-----100 PERCENT TORQUE
C
      QRAFCT=100.0/(COLMAXT-COLMINT)
C
C
C-----MULTIPIER INSTEAD OF DIVISOR
C
      C1687=1.0/1687.78
C-----INITIAL PITCH ATTITUDE
C
      THETIC = ( XUT*UB/G ) * R2D + THETOFF
      THETR  = THETIC * D2R
C
C
C-----EQUATE GAMH AND PSI.
C
      GAMHIC = PSIIC
C-----CALC. INITIAL PILOT CONTROL VALUES FOR ZEROING IN OP.
C     IF ICOZERO = 1, OTHERWISE USE ACTUAL VALUES
      ROLLOIC  = 0.
      PITCHIC = 0.
      YAWOIC   = 0.
      IF ( ICOZERO .EQ. 0 ) GOTO 90
          ROLLOIC = ROLLOP
          PITCHIC= PITCHOP
          YAWOIC   = YAWOP
   90 CONTINUE
C
C
C-----RESET LOADER DRIVES AFTER USING MAG BRAKE.
C
      TRIM1C = ROLLOIC
      TRIM2C = PITCHIC
      TRIM3C = YAWOIC
C
C
C-----MAG BRAKE STATUS EQUALS MOMENTARY SWITCH POSITION IN
C
      IMB = IMBCP + IMBC !this statement was IMB=IMBCP=IMBC
C                          failed on compile. GLH 5/21/87
C
```

```
C
C
C----- I. C. VALUE OF SGVOL.
C
      TINC = TINCIC
      SGVOL=SGVOLIC
      GO TO 101
  100 CONTINUE
C
C***********************************************************
C*****************    OPERATE  SECTION    *****************
C***********************************************************
C
C------OPERATE VALUE OF SOUND GENERATOR VOLUME
C
      TINC = TINCOP
      SGVOL=SGVOLOP
  101 CONTINUE
C
C
C-----CONVERT LONGITUDINAL BODY VELOCITY TO KNOTS.
C
      UBKTS = UB*FPS2KOS
C
C
C
C----------   COLLECTIVE INPUT    --------------
C------------------------------------------------
C
C
C-----COLLECTIVE TRIM
C-----SLOPE AND INTERCEPT ARE FUNCTION OF AIRSPEED
C
C
      IF( UBKTS .LE.   0.0 ) GO TO 240
      IF( UBKTS .LE.  40.0 ) GO TO 250
      IF( UBKTS .LE.  90.0 ) GO TO 260
      IF( UBKTS .LE.140.0 ) GO TO 270
      GO TO 280
C
C----------- HOVER
  240 CONTINUE
      SLP   = SLP1
      YINT  = YINT1
      GO TO 290
C
C----------- 0 TO 40 KNOTS
  250 CONTINUE
      SLP   = SLP2
      YINT  = YINT2
      GO TO 290
C
C----------- 40 TO 90 KNOTS
  260 CONTINUE
      SLP   = SLP3
```

```
      YINT   = YINT3
      GO TO 290
C
C----------- 90 TO 140 KNOTS
  270 CONTINUE
      SLP    = SLP4
      YINT   = YINT4
      GO TO 290
C
C----------- ABOVE 140 KNOTS
  280 CONTINUE
      SLP    = SLP5
      YINT   = YINT5
C
C
C-----COLLECTIVE TRIM, 0(DOWN) TO 10(UP)
C
  290 CONTINUE
      COLTRM = (YINT + SLP*UBKTS)*0.1
C
C...RATE OF DESCENT LIMIT.
C
      IF(UBKTS .GE. 70.0)GO TO 300
C
      RODLIM=(RODINT1 + RODSLP*UBKTS)*0.1
      GO TO 305
C
  300 CONTINUE
C
      RODLIM=(RODINT2 - RODSLP*UBKTS)*0.1
C
  305 CONTINUE
C
C
C
C...AUTOMATICALLY TRIM IN I. C. E.G. FOR DYNAMIC CHECKS.
C
C
      IF((IMODE .LT. 0) .AND. (LTRIM .NE. 0))COLOP=COLTRM
C
C
C
C-----FILTER COLLECTIVE INPUT TO SIMULATE
C-----ENGINE RESPONSE
C
      COLOZ = COLF1*COLOZ + COLF2*COLOP
C
C
C-----CLIMB/DIVE COMMAND (PILOT - TRIM)
C
      EROM4 = COLOZ - COLTRM
C
C
C-----MAX. RATES OF CLIMB AND DESCENT LIMITATIONS.
C
```

```
      IF(UBKTS .GT. 70.0) GO TO 310
            ROCLIM = 0.1*(XINT1(ICON) + RSLP1(ICON)*UBKTS)
            GO TO 320
  310 CONTINUE
      ROCLIM = 0.1*(XINT2(ICON) + RSLP2(ICON)*UBKTS)
  320 CONTINUE
      IF(EROM4 .GT. ROCLIM) EROM4 = ROCLIM
      IF(EROM4 .LT. RODLIM) EROM4=RODLIM
C
C
C-----ENGINE TORQUE FOR INSTRUMENT
C
      LIMRPM = 0
      QRAPCT = QRAFCT*(COLOZ - COLMINT) + PCTBIAS
      IF( QRAPCT.GE.98.) LIMRPM=1
C
C--------------------------------------------------
C---------   STICK AND PEDAL INPUTS   -------------
C
      EROM1 = ROLLOP  - ROLLOIC
      EROM2 = PITCHOP - PITCHIC
      EROM3 = YAWOP   - YAWOIC
C
C
C-----GAIN PILOT COMMANDS
C
      ROLLOZ  = OOOG1*EROM1
      PITCHOZ = OOOG2*EROM2
      YAWO    = OOOG3*EROM3
      COLO    = OOOG4*EROM4 + OOOB4
C
C
C
C-----DEADBAND INPUTS FROM PILOT/AUTOPATH
C
      CALL DEAD(   ROLLOZ,   RDBO )  !COMMENTED FOR DEBUG SUSP
      CALL DEAD( PITCHOZ,  PDBO )
      CALL DEAD(     YAWO,  YDBO )
C      CALL DEAD(     COLO, CDBO )
C
C
C-----LIMIT PILOT/AUTOPATH INPUTS
C
      IF( ROLLOZ   .LT. -ROLMAX ) ROLLOZ  = -ROLMAX
      IF( ROLLOZ   .GT.  ROLMAX ) ROLLOZ  =  ROLMAX
      IF( PITCHOZ  .LT. -PITMAX ) PITCHOZ = -PITMAX
      IF( PITCHOZ  .GT.  PITMAX ) PITCHOZ =  PITMAX
      IF( YAWO     .LT. -YAWMAX ) YAWO    = -YAWMAX
      IF( YAWO     .GT.  YAWMAX ) YAWO    =  YAWMAX

      IF( COLO     .LT. COLOMIN ) COLO    = COLOMIN
      IF( COLO     .GT. COLOMAX ) COLO    = COLOMAX


C
```

```
C-------------------------------------------------
C---------------   GAIN SCEDULES   ---------------
C
C
C-----DIRECTIONAL GAINS, WEATHER VANE EFFECT
C
      IF(VEQ .GT. 30.0) GO TO 350
           XNVO   = 0.0
           ZNPO   = 0.0
           XNPHIO = 0.0
           GO TO 370
  350 CONTINUE
      IF(VEQ .GT. 50.0) GO TO 360
           XNVO   = XNV*VEQ*C1687 - 0.02365
           ZNPO   = 0.01908*VEQ   - 0.57235
           XNPHIO= -ZNPO*ZNRT
           GO TO 370
  360 CONTINUE
      IF(UB .LT. 0.00006) UB=0.001
      XNVO   = XNV/UB
      ZNPO   = G/UB
      XNPHIO =-ZNPO*ZNRT
  370 CONTINUE
C
C
C-----VEHICLE SENSITIVITIES DUE TO TURBULENCE
C
CC      CALL VBARG(VEQ,VEQRT)! COMMENTED BY GLH 6/16/87
C
CC      ZLVT = FIXGN1(ZLVMT) ! COMMENTED BY GLH 6/16/87
CC      ZMUT = FIXGN1(ZMUMT) ! COMMENTED BY GLH 6/16/87
CC      ZMWT = FIXGN1(ZMWMT) ! COMMENTED BY GLH 6/16/87
CC      ZNVT = FIXGN1(ZNVMT) ! COMMENTED BY GLH 6/16/87
CC      ZUT  = FIXGN1(ZUTMT) ! COMMENTED BY GLH 6/16/87
CC      ZWTT = FIXGN1(ZWTMT) ! COMMENTED BY GLH 6/16/87
C
C
C-----SIDESLIP LIMIT

      IF(VEQ .GT. VEQPT(ICON)) GO TO 420
            BETMAX = ESLP1(ICON)*VEQ + EINT1(ICON)

            GO TO 430
  420 CONTINUE
      BETMAX = ESLP2(ICON)*VEQ + EINT2(ICON)
  430 CONTINUE
C
C-----HUD VARIABLE
C
      BETATMP = BETA/BETMAX
      BETAHUD = BETATMP*BETAFG
      IF( VEQ .LT. 30. )BETAHUD=BETAHUD*VEQ*.03333
      LIMBET  = 0
      IF( ABS(BETATMP) .GE. .95 ) LIMBET=1
      IF( VEQ .LT. 30. ) LIMBET=0
```

```
C
C
C-----MAX EFFECTIVE PEDAL DEFLECTION
C
      PEDMAX = -BETMAX*SLPFACT*D2R
      IF(YAWO  .GT.  PEDMAX) YAWO =  PEDMAX
      IF(YAWO  .LT. -PEDMAX) YAWO = -PEDMAX
C
C-----CONTROL SYSTEM TYPE
C
      GO TO (500,540,580)     ICS
C
C-----------------------------------------------------
C--------    PURE RATE COMMAND SYSTEM:ICS=1    ------
C
C
  500 CONTINUE
C
      ROLLO     = ROLLOZ
      PITCHO    = PITCHOZ
      XLPHIO    = XLPHI1
      XMTHETO   = XLPHIO
      ZMQT      = ZMQ1
      ZLPT      = ZLP1
      XMDET     = XMDE1
      XLDAT     = XLDA1


      GO TO [illegible]
C
C
C-----------          -----  ----------------- ------
C--------  ATT. [illegible]    ATT. HOLD. ICS=2   --- ----
C
C
  540 CONTINUE
C
      ROLLO     = ROLLOZ
      PITCHO    = PITCHOZ
      XLPHI     [illegible]
      ZMQT      ZMQ1
      XMTHETO   XLPHIO
      ZLPT      ZLP1
C
      IF(VEQ [illegible]) GO TO 550
          XMDET  [illegible]
          XLDAT  [illegible]
          GO TO 660
  550 CONTINUE
      IF(VEQ [illegible] TO [illegible]
          [illegible]          0.006 [illegible]*VEQ
          [illegible]          0.0025*VEQ
          [illegible]
  [illegible]
```

```
      XMDET = 0.13
      GO TO 600
C
C----------------------------------------
C-----   RATE COMM. / ATT. HOLD. ICS=3
C
C
  580 CONTINUE
C
C-----PITCH AXIS.
C       CALL UPDATE(PITCHOZ,2,IMODE,BUFFP,XXP)!COMMENTED OUT
C
      PITCHO=BBP(1)*XXP(1) + BBP(2)*XXP(2) +
     .       BBP(3)*(PITCHOZ - AAP(1)*XXP(1) - AAP(2)*XXP(2)
C
C-----ROLL AXIS.
C       CALL UPDATE(ROLLOZ,2,IMODE,BUFFR,XXR)!COMMENTED OUT 6
C
      ROLLO =BBR(1)*XXR(1) + BBR(2)*XXR(2) +
     .       BBR(3)*(ROLLOZ - AAR(1)*XXR(1) - AAR(2)*XXR(2))
C
      XLPHIO  = XLPHI3
      XMTHETO = XLPHIO
      ZMQT    = ZMQ1
      ZLPT    = ZLP1
      XLDAT   = XLDA3
      XMDET   = XMDE3
  600 CONTINUE

 9999 RETURN
      END
```

SMART.FOR

```
C**********  SOURCE FILE NAME - SSMARTM
C
C
C
C*****************************************************
C*********                 HAC3 SSMARTM              *****
C*****************************************************
C
C
C
C     TITLE     SMART (CDC 7600 VERSION) - UPDATED 9/17/84

C23456789012345678901234567890123456789012345678901234567
C        1         2         3         4         5       6       C
C
C
      SUBROUTINE SMART
C
C
C
C*******************************************************
C                    SUBROUTINE ABSTRACT & HIERARCHY
C*******************************************************
C
C                               ( CDC 7600 VERSION )
C
C                    SMART IS A FAST LOOP ROUTINE WHICH
C                SOLVES THE STANDARD KINEMATICAL EQUATIONS
C                AND PROVIDES A STANDARD ATMOSPHERE MODEL.
C                IT IS CALLED BY FASTP AND IN TURN CALLS ARDC
C
C
C
C********************************************************
C                    CREATION & MODIFICATION LOG
C********************************************************
C
C
C
C
C
C     R. E. MCFARLAND -CSC- MARCH 1976
C
C     11/15/84              MODIFIED TO MEET BASIC NOTE #123
C
C     08/13/84  B.CHUNG     MODS TO MAKE PILOT EYEPOINT DIFF
C                           FROM MOTION POINT BY ZPE ( H DIS
C                           FROM PILOT MOTION POINT TO PILOT
C                           POINT)
C
C     09/17/85  P.RANDALL  SYRE  REFORMATTED
C
```

```
C      02/28/85  SCB         REMOVED SMART TIMESAVER FOR CDC
C
C      12/29/85  SCB         MODS TO REMOVE ANY COMBO OF DEGR
C                            FREEDOM WITH FLAGS : IPON TO IVD
C
C      NOV 1986             SET UP FOR HAC - III SIMULATION:
C                           REMOVAL OF CALCULATIONS DONE IN C
C
C      JUN 1987             Replaced COMMON, EQUIVALENCE, AND
C                           STATEMENTS with INCLUDE HAC.INC.
C
C      01/19/1988 MCL       FIXED NEGATIVE ALTITUDE.
C
C
C
C
C******************************************************
C
              INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
           INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'
              INCLUDE '[HELHAC.FOR]HAC3.INC/LIST'
           INCLUDE '[HELHAC.FOR]FLTDATA.INC/LIST'
C
C******************************************************
C                  E X E C U T A B L E   C O D E
C******************************************************
C
C
C       DELT = DT2
C
C
C
C*****TOTAL AIRCRAFT FORCE COMPONENTS
C
C      FTX=FAX+FEX+FGX
C      FTY=FAY+FEY+FGY
C      FTZ=FAZ+FEZ+FGZ
C
C...THE ABOVE CALCULATIONS ARE DONE IN CONTR2.
C
C
C
C
C*****TOTAL AIRCRAFT MOMENT COMPONENTS
C
C      TTL=TAL+TEL+TGL
C      TTM=TAM+TEM+TGM
C      TTN=TAN+TEN+TGN
C
C
C*****TOTAL AIRCRAFT FORCE COMPONENTS IN EARTH FRAME
C
      FN=T11*FTX+T21*FTY+T31*FTZ
      FE=T12*FTX+T22*FTY+T32*FTZ
      FD=T13*FTX+T23*FTY+T33*FTZ
```

```
C
C
C
C*****DOWNWARD ACCELERATION
C
      TEMP=1./XMASS
      VDD=(FD+FG)*TEMP
C
C
C*****DISTANCE FROM EARTH'S CENTER TO AIRCRAFT
C
      RTV=ALT+RE
C
C
      IF(IFLAT) 420,415,420
C
C
C*****AIRCRAFT ACCELERATION COMPONENTS FROM ROTATING
c EARTH FRAME
C
 415  CONTINUE
      RTVINV=1.0/RTV
      TEMP1=RTVINV
      VND=FN*TEMP+(VN*VD-VE**2*TLAT)*TEMP1
      VED=FE*TEMP+(VE*VD+VN*VE*TLAT)*TEMP1
      VDD=-(VE**2+VN**2)*TEMP1+VDD
      RCLAT=RTV*CLAT
      ROUNDV=-OMEG*RCLAT
      GO TO 430
C
C
C*****AIRCRAFT ACCELERATION COMPONENT FROM FIXED,
cFLAT EARTH FRAME
C
 420  VND=FN*TEMP
      VED=FE*TEMP
      TEMP1=0.
      ROUNDV=0.
      RCLAT=RE
      REINV=1./RE
      RTVINV=REINV
C
C
 430  IF(IMODE) 440,1000,450
C
C
C*******************************************************
C                  I N I T I A L   C O N D I T I O N S
c                 ( I. C .)
C*******************************************************
C
C
C
 440  DTIME=0.
      DELTI=0.
```

```
      DELTH=0.
      DTINV=0.
      GO TO 465
C
C
C
C*******************************************************
C                  O P E R A T E ( OP )
C*******************************************************
C
C
C
 450  DELTI=0.5*DELT
C
C
C*****ADDED FEB.77 BY R.E.MCFARLAND
C*****IFREEZ=1 FIXES TRANSLATIONAL AS WELL AS ROTATIONAL
c     MOTION
C*****(I.E. IFFCI IS IMMATERIAL IF IFREEZ=1)
C
      IF(IFREEZ.EQ.1) DELTI=0.
      DTINV=1.0/DELT
      IF(IFFCI) 460,460,455
 455  DELTH=0.
      GO TO 462
C
 460  DELTH=DELTI
 462  DTIME=DTIME+DELT
 465  TIME=DTIME
C
C
C
C******************************************************
C*******     AIRCRAFT  LINEAR  ACCELERATIONS       *****
C******************************************************
C
C
C
C
C--------------------
C-----  ARTI ARMCOP MODIFICATIONS
C--------------------
C
C*****INTEGRATE THE EARTH FRAME ACCELERATION COMPONENTS
c     INTO
C*****EARTH FRAME VELOCITIES USING AN ADAMS' SECOND-ORDER
C*****PREDICTOR FORMULA.
C...Added to freeze airspeed when needed.  MCL 01/25/88
      AIR_DELTH = DELTH
      IF (AIRSPEED_HOLD .EQ. .TRUE.) THEN AIR_DELTH = 0.0
c      VN=VN+DELTH*(3.0*VND-VNDP)
C      VE=VE+DELTH*(3.0*VED-VEDP)
C      VD=VD+DELTH*(3.0*VDD-VDDP)
C.........
      VN=VN+AIR_DELTH*(3.0*VND-VNDP)
```

```
      VNDP=VND
      VE=VE+AIR_DELTH*(3.0*VED-VEDP)
      VEDP=VED
      VD=VD+AIR_DELTH*(3.0*VDD-VDDP)
      VDDP=VDD
C
C
C*****CREATE AIRCRAFT VELOCITIES IN LATITUDE AND LONGITUDE
c     AXES.
C
      VEE=VE+ROUNDV
      XLATD=VN*RTVINV
      XLOND=VEE/RCLAT
C
C
C*****INTEGRATE THE VELOCITIES IN THE LAT/LONG AXES INTO
C*****LATITUDE AND LONGITUDE USING A TRAPEZOIDAL CORRECTOR
C*****.'ORMULA.
C
      XLON=XLON+DELTH*(XLOND+XLONDP)
      XLONDP=XLOND
      XLAT=XLAT+DELTH*(XLATD+XLATDP)
      XLATDP=XLATD

      ALT_DELTH = DELTH
      IF (ALT_HOLD .EQ. .TRUE.)ALT_DELTH = 0.0
c
      ALT=ALT-ALT_DELTH*(VD+VDP)
      IF (ALT .LT. 0.0) ALT = 0.0
C
      VDP=VD

C
C*****AIRCRAFT POSITION RELATIVE TO RUNWAY
C
      ALTD=-VD
      HCG=ALT-HR
      DNR=RR*(XLAT-XLATR)
      DER=RR*CLATR*(XLON-XLONR)

C
C
C*****************************************************
C
C
C
C
C
C--------------------
C
C...THESE CALCULATIONS ARE DONE IN CONTR2.
C
C
C
C     PBD=IPON*((XMC(1)*RB+XMC(2)*PB)*QB+XMC(3)*TTL+XMC(4)
```

```
C     1           *TTN + XMCC1*QB + XMCC5*RB - XMCC6*PB)
C      QBD=IQON*(XMC(5)*RB*PB+XMC(6)*(RB**2-PB**2)+XMC(7)
C     1           *TTM - XMCC2*RB + XMCC3*PB)
C      RBD=IRON*((XMC(8)*PB+XMC(9)*RB)*QB+XMC(4)
      1           *TTL+XMC(10)*TTN
C     1           + XMCC4*QB + XMCC6*RB - XMCC7*PB)
C
C
C*****INTEGRATE THE BODY AXES ANGULAR ACCELERATIONS INTO
c*****BODY
C*****AXES ANGULAR VELOCITIES USING AN ADAMS' SECOND-ORDER
C*****PREDICTOR FORMULA.
C
      PB=PB+DELTI*(3.0*PBD-PBDP)
      PBDP=PBD
      QB=QB+DELTI*(3.0*QBD-QBDP)
      QBDP=QBD
      RB=RB+DELTI*(3.0*RBD-RBDP)
      RBDP=RBD
C
C
C*****AIRCRAFT TOTAL ROTATIONAL RATE, EARTH FRAME
C
      PT=PB-PLB
      QT=QB-QLB
      RT=RB-RLB
C
C
C*****AIRCRAFT TOTAL ROTATIONAL RATES, BODY FRAME
C
C     PBWN=PB+PTURB
C     QBWN=QB+QTURB
C     RBWN=RB+RTURB
C
C...THE ABOVE CALCULATIONS ARE COMMENTED OUT BECAUSE
c   TURBULENCE FACTORS
C... APPEAR IN CONTR2.
C
C
      PBWN=PB
      QBWN=QB
      RBWN=RB
C
C
C*****AIRCRAFT ROTATIONAL RATES, EARTH FRAME
C
      THED=QT*CPHI-RT*SPHI
      PSID=(QT*SPHI+RT*CPHI)/CTHT
      PHID=PT+PSID*STHT
C
C
C*****INTEGRATE THE EARTH FRAME ANGULAR VELOCTIES INTO
c     EULER
C*****ANGLES USING A TRAPEZOIDAL CORRECTOR FORMULA.
C
```

```
C...Added to hold heading.  MCL 01/26/1988
      HEADING_DELTI = DELTI
      IF (HEADING_HOLD .EQ. .TRUE.) HEADING_DELTI =0.0
c...
       PSIR_P = PSIR
       THETR=THETR+DELTI*(THED+THEDP)
       PSIR=PSIR+HEADING_DELTI*(PSID+PSIDP)
       PHIR=PHIR+DELTI*(PHID+PHIDP)
       THEDP=THED
       PSIDP=PSID
       PHIDP=PHID
C
C
C*****NEW SINE & COSINE OF EULER ANGLES
C
       SPHI=SIN(PHIR)
       CPHI=COS(PHIR)
       SPSI=SIN(PSIR)
       CPSI=COS(PSIR)
       STHT=SIN(THETR)
       CTHT=COS(THETR)
C
C
C*****NEW TRIGONOMETRIC MULTIPLIERS
C
       T11=CTHT*CPSI
       T21=SPHI*STHT*CPSI-CPHI*SPSI
       T31=CPHI*STHT*CPSI+SPHI*SPSI
       T12=CTHT*SPSI
       T22=SPHI*STHT*SPSI+CPHI*CPSI
       T32=CPHI*STHT*SPSI-SPHI*CPSI
       T13=-STHT
       T23=SPHI*CTHT
       T33=CPHI*CTHT
C
C**************************************************************
C
C
C
C*****PILOT EYEPOINT POSITION RELATIVE TO RUNWAY IN EARTH
c      AXES
C
       DNPR=DNR+T11*XP+T21*YP+T31*(ZP-ZPE)
       DEPR=DER+T12*XP+T22*YP+T32*(ZP-ZPE)
C
       XPR=DNPR*CTHETR+DEPR*STHETR
       YPR=-DNPR*STHETR+DEPR*CTHETR
       HPR=HCG-T13*XP-T23*YP-T33*(ZP-ZPE)

c     WRITE(9,1999)DNR,DER
c         WRITE(9,2000)XPR,YPR
c 1999    FORMAT(1X,'DNR = ',F18.4,' DER = ',F18.4)
c 2000    FORMAT(1X,'XPR = ',F18.4,' YPR = ',F18.4)
C
C
```

```
C*****AIRCRAFT C.G. RELATIVE TO RUNWAY IN RUNWAY AXES
C
      XCG=DNR*CTHETR+DER*STHETR
      YCG=-DNR*STHETR+DER*CTHETR
C
C
C*****TURBULENCE VELOCITY COMPONENTS
C
C*****IF IETURB.EQ.1 THE RANDOM TURBULENCE IS ALREADY
c     GENERATED IN THE
C*****EARTH AXES AS VNTURB, VETURB, VDTURB. OTHERWISE,
c     THESE EQUATIONS
C*****EXPECT UTURB, VTURB AND WTURB.

      IF(IETURB.EQ.0) GO TO 500
      UTURB=T11*VNTURB+T12*VETURB+T13*VDTURB
      VTURB=T21*VNTURB+T22*VETURB+T23*VDTURB
      WTURB=T31*VNTURB+T32*VETURB+T33*VDTURB
      GO TO 520
C
 500  VNTURB=T11*UTURB+T21*VTURB+T31*WTURB
      VETURB=T12*UTURB+T22*VTURB+T32*WTURB
      VDTURB=T13*UTURB+T23*VTURB+T33*WTURB
C
C
  520 CONTINUE
C
C
C
C*****TOTAL WIND VELOCITIES
C
C     VTWN=VNW+VNTURB
C     VTWE=VEW+VETURB
C     VTWD=VDW+VDTURB
C
C...ABOVE CALCULATIONS COMMENTED OUT BECAUSE TURBULENCE
c   EFFECTS ARE
C...TAKEN INTO ACCOUNT IN CONTR2.

      VTWN=VNW
      VTWE=VEW
      VTWD=VDW
C
C
C
C*****AIRCRAFT RELATIVE VELOCITIES
C
      VNR=VN-VTWN
      VER=VEE-VTWE
      VDR=VD-VTWD
C
C
C*****AIRCRAFT VELOCITY (BODY FRAME)
C
      UB=T11*VNR+T12*VER+T13*VDR
```

```
      VB=T21*VNR+T22*VER+T23*VDR
      WB=T31*VNR+T32*VER+T33*VDR
C
C*****ANGLE OF ATTACK W/ SINE AND COSINE
C
      IF(ABS(UB).LT.0.00001) UB=SIGN(0.00001,UB)
      ALFAR=ATAN2(WB,UB)
      SALPH=SIN(ALFAR)
      CALPH=COS(ALFAR)
C
C
C*****SIDESLIP ANGLE W/SINE AND COSINE
C
      DUM2=UB**2+WB**2
      DUM=SQRT(DUM2)
      DUM1=SIGN(DUM,UB)
      BETAR=ATAN2(VB,DUM1)
      SBETA=SIN(BETAR)
      CBETA=COS(BETAR)
C
C
C*****AIRCRAFT VELOCITY WITH RESPECT TO WIND
C
      VRW2=DUM2+VB**2
      VRW=SQRT(VRW2)
C
C
C*****AIRCRAFT ACCELERATION (BODY AXIS)
C*****WIND ACCELERATION INCLUDES ONLY MEAN WIND
C
      ATWN=(VNW-VNWP)*DTINV
      ATWE=(VEW-VEWP)*DTINV
      ATWD=(VDW-VDWP)*DTINV
      VNWP=VNW
      VEWP=VEW
      VDWP=VDW
C
      RAN=VND-ATWN
      RAE=VED-ATWE
      RAD=VDD-ATWD
      UBD=RT*VB-QT*WB+T11*RAN+T12*RAE+T13*RAD
      VBD=PT*WB-RT*UB+T21*RAN+T22*RAE+T23*RAD
      WBD=QT*UB-PT*VB+T31*RAN+T32*RAE+T33*RAD
C

C
      IF(IMODE) 632,636,636
C
C
C
C*******************************************************
C                  I N I T I A L    C O N D I T I O N S
c                 ( I. C .)
C*******************************************************
C
```

```
C
C
 632  IF(ITOMTR) 636,636,634
 634  ALFD=0.
      BETD=0.
       GO TO 640
C
C
C
C*****************************************************
C                    O P E R A T E ( OP )
C*****************************************************
C
C
C
 636  CONTINUE
      ALFD=(UB*WBD-WB*UBD)/DUM2
      BETD=(DUM2*VBD-(UB*UBD+WB*WBD)*VB)/(DUM1*VRW2)
 640  CONTINUE
C
C
C*****AIRCRAFT ACCELERATION, BODY FRAME
C
      AX=FTX*TEMP
      AY=FTY*TEMP
      AZ=FTZ*TEMP
      ANZ=-(AZ/G)
C
C
C*****PILOT ACCERATION, BODY FRAME
C
     AXP=AX-(RB**2+QB**2)*XP+(PB*QB-RBD)*YP+(PB*RB+QBD)*ZP
     AYP=AY+(PB*QB+RBD)*XP-(RB**2+PB**2)*YP+(QB*RB-PBD)*ZP
     AZP=AZ+(PB*RB-QBD)*XP+(QB*RB+PBD)*YP-(QB**2+PB**2)*ZP
C
C*****ALPHA, BETA, & EULER ANGLES IN DEGREES
C
      THET=THETR*R2D
      PHI=PHIR*R2D
      PHIRDM=PHI
      PSI=PSIR*R2D
      ALFA=ALFAR*R2D
      BETA=BETAR*R2D
C
C
C*****EQUIVALENT AIRSPEED
C
      VEQ=TEMA*VRW
C
C*****INSTANTANEOUS ROTATIONAL RATES (L-FRAME)
C
      PL=VE*TEMP1
      QL=-VN*TEMP1
      RL=-PL*TLAT
C Turn_rate ADDED 7/21/88 GLH. COMPUTES TURN RATE IN
```

```
c  RADIANS/SEC.

      IF(DELTI .NE. 0.) TURN_RATE = (PSIR - PSIR_P)/DELTI
C
C*****INSTANTANEOUS ROTATIONAL RATES (BODY FRAME)
C
      PLB=T11*PL+T12*QL+T13*RL
      QLB=T21*PL+T22*QL+T23*RL
      RLB=T31*PL+T32*QL+T33*RL
C
C
C*****AIRCRAFT TOTAL VELOCITY & GROUND SPEED
C
      PAD=VN**2+VEE**2
      VT=SQRT(PAD+VD**2)
      VG=SQRT(PAD)
C
C
C*****FLIGHT PATH ANGLES
C
      IF(VT.EQ.0.0) GO TO 735
      GAMV=ATAN2(-VD,VG)
      IF(PAD.EQ.0.0) GO TO 735
      GAMH=ATAN2(VEE,VN)
 735  CONTINUE
C
C
C*****ACCELERATION DUE TO GRAVITY
C
      FG=WAIT*(RE*RTVINV)**2
      G=FG*TEMP

C
C
C*****LATITUDE TRIGONOMERIC MULTIPLIERS
C
      SLAT=SIN(XLAT)
      CLAT=COS(XLAT)
      TLAT=SLAT/CLAT
C
C
C*****AIR DENSITY AND SPEED OF SOUND
C
      IF(ICOND) 804,804,806
 804  XX=ALT
      GO TO 807
C
 806  XX=HRHOZ
 807  CONTINUE
C       CALL ARDC62(XX,SOUND,RHO)!COMMENTED OUT
c       ARDC 6/19/87 GLH
      RHO=RHO/YCON
      SOUND=SOUND*SQRT(YCON)
C
C*****TEMPERATURES, PRESSURES, AND THEIR RATIOS
```

```
C
      TR=1.+.2*XMACH**2
      IF(XMACH-XM1) 810,810,820
 810  PR=TR*TR*TR*SQRT(TR)
      GO TO 830
C
 820  IF(XMACH) 825,825,828
 825  PR=1.0
      GO TO 830
C
 828  CONTINUE
      TEMPA=XMACH**2
      TEMP2=7.-1./TEMPA
      PR=166.9*TEMPA/(TEMP2*TEMP2*SQRT(TEMP2))
C
 830  IF(ALT-HLEV) 840,840,850
 840  TAMBR=1.-6.875E-6*ALT
      PAMBR=TAMBR**5.256
      GO TO 860
C
 850  TAMBR=.751895
      PAMBR=.2234*EXP(-4.806E-5*(ALT-HLEV))
 860  CONTINUE
      TAMB=DELAT+TAMBR*TEMPZ*.5555555
      PAMB=PAMBR*PRESZ
      PTOT=PR*PAMB
      TTOT=TR*TAMB
      QBARC=PTOT-PAMB
C
C
C*****CALIBRATED AIRSPEED
C
C*****MODIFY RHO AND SOUND BY DELTA TEMPERATURE EFFECT
      YCON=TAMB/(TAMB-DELAT)
      XMACH=VRW/SOUND
      RHO2=0.5*RHO
      TEMA=.59249*SQRT(RHO/RHOZ)
      IF(VEQ-10.0) 870,880,880
 870  VCAL=VEQ
      GO TO 890
C
 880  VCAL=.59249*SOUNDZ*SQRT(5.0*
     1      ((QBARC/PRESZ+1.0)**.2857-1.0))
 890  CONTINUE
C
C
 1000 CONTINUE
C
C
      QBAR=RHO2*VRW2
C
C
C
      RETURN
      END
```

FAM.FOR

```
C     TITLE FAM
C
C
      SUBROUTINE FAM
C
C
C
C*****************************************************************
C************    SUBROUTINE ABSTRACT AND HIERARCHY   *************
C*****************************************************************
C
C
C   ROUTINE PROVIDES A SIMPLE HELICOPTER INCLUDING
C   AERO, ENGINE, AND PART OF THE EQUATIONS OF MOTION.
C   IT REQUIRES A MODIFIED SMART.
C
C
C*****************************************************************
C************       CREATION AND MODIFICATION LOG   *************
C*****************************************************************
C
C
C 6/25/87 DERIVED FROM CONTR2.FOR, GLH
C*****************************************************************
C*************        SIGNIFICANT VARIABLES    *****************
C
C
C----------------
C-- INPUTS
C----------------
C
C ICONFI    CRAFT CONFIGURATION:  UH60, AH1S OR LHX.
C
C
C APPPCT      APPROACH PERCENT, 0-100
C             PERCENT OF FULL ENGINE POWER FOR YELLOW LIGHT
C REDTIME     TIME AT 100 PERCENT TORQUE FOR RED LIGHT
C
C
C-----------------
C--  OUTPUTS
C-----------------
C
C FZMAX       MAX Z FORCE FOR CURRENT AIRSPEED
C FZMIN       MIN
C APPMAX      MAX Z FORCE APPROACH LIMIT FOR CURRENT AIRSPEED
C APPMIN      MIN
C KYELLOW     Z FORCE APPROACH LIMIT REACHED      0 OR 1
C KRED        Z FORCE LIMIT REACHED               0 OR 1
C******************************************************************
C
```

```
                INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
                 INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'

C*****************************************************************
C*****************   EXECUTABLE  CODE   **************************
C*****************************************************************
C
C-----CALCULATE PITCH ANGLE DUE TO VELOCITY
C
      TMDUM=-XUT*UB/G
C
C
C-----------------------------------------------------------
C---------    TARGET ROTATIONAL ACCELS. (RADS/S**2)  ----------
C
C
      PBD = ZLPT*PB + XLDAT*ROLLO  + XLPHIO*PHIR
     .    + ZLV *VB + ZLVT *VTURB + PBDR
C
      QBD = ZMQT*QB + XMDET*PITCHO + XMTHETO*(THETR  + TMDUM)
     .    + ZMU *UB + ZMUT *UTURB  + ZMWT*WTURB + QBDR

C
      RBD = ZNRT*RB + XNDPT*YAWO   + XNPHIO*PHIR
     .    + XNVO*VB + ZNVT *VTURB  + ZNPO*PB + RBDR

C------------------------------------------------------------
C--------- TOTAL TRANSLATIONAL FORCES ON VEHICLE  ------------
C
C
      FTX  = XMASS*( XRT + XUT*(UB +     UTURB) ) + FRX
      FTY  = XMASS*( YRT + YVT*(VB +     VTURB) ) + FRY
      FTZ  = XMASS*( ZRT + ZWT* WB + ZUT*UTURB + ZWTT*WTURB
          +  ZDCT*COLO ) + FRZ

C
C
C
C----------------------------------------
C--------  FTZ LIMIT CALCULATIONS  --------
C
C
C------GET APPROPRIATE SLOPE AND INTERCEPT
C------FOR MIN/MAX LOAD FACTOR
C
      DO 700 MM=1,5
      IF(VEQ .GT. VQMAX(MM+1)) GO TO 700
          SLPMAX=SLPMX(MM)
          YINTMX=RYINT(MM)
          GO TO 710
  700 CONTINUE
  710 CONTINUE
C
```

```
C
      DO 740 JJ=1,5
      IF(VEQ .GT. VQMIN(JJ+1)) GO TO 740
           SLPMIN=SLPMN(JJ)
           YINTMN=YMINT(JJ)
           GO TO 750
  740 CONTINUE
  750 CONTINUE
C
C
C------CALCULATE MIN / MAX FOR FTZ
C------(NOTE: FZ IS POSITIVE DOWN] ...JWB)
C
      ZNMAX = YINTMX + SLPMAX*VEQ
      ZNMIN = YINTMN + SLPMIN*VEQ
C
      FZMIN = -ZNMAX * WAIT
      FZMAX = -ZNMIN * WAIT
C
C
C-----CALCULATE FTZ APPROACH LIMITS.
C
      APPMIN=FZMIN - (FZMIN - WAIT)*FAPPCT
      APPMAX=FZMAX + (FZMAX - WAIT)*FAPPCT
C
C
C
C------------------------------------------
C--------  FTZ LIMIT APPLICATION   --------
C
      IF( FTZ .LT. FZMIN ) FTZ=FZMIN
      IF( FTZ .GT. FZMAX ) FTZ=FZMAX
c    WRITE(6,200)FZMIN,FZMAX
c200          FORMAT(1X,'FZMIN=',F14.4,' FZMAX=',F14.4)
C
      LIMG  = 0
      IF( (FTZ.LE.(.95*FZMIN)) .OR. (FTZ.GE.(.95*FZMAX)))
     ,     LIMG=1
     RETURN
     END
```

BEEPER.FOR

```
C     TITLE                                           BEEPER
C
C
      SUBROUTINE BEEPER
C
C
C
C*******************************************************
C************    SUBROUTINE ABSTRACT AND HIERARCHY  *****
C*******************************************************
C
C
C   ROUTINE PROVIDES FOR TRIM BEEPER AND MAG BRAKE.
C   IT REQUIRES A MODIFIED SMART.
C
C
C*******************************************************
C************       CREATION AND MODIFICATION LOG   *****
C*******************************************************
C
C
C 6/25/87 DERIVED FROM CONTR2.FOR, GLH
C*******************************************************
C*************        SIGNIFICANT VARIABLES    **********
C
C
C
C  ROLLOP     :LATERAL CYCLIC         +- 6   INCHES
C  PITCHOP    :LONGITUDINAL CYCLIC    +- 6   INCHES
C  YAWOP      :PEDAL                  +- 3.5 INCHES
C  COLOP      :COLLECTIVE             0 - 10 INCHES
C
C IMBC        MAG BRAKE MOMENTARY FROM CAB      0 OR 1
C IRWD        TRIMM BEEPER DISCRETES FROM CAB   0 OR 1
C ILWD
C IANU
C IAND
C
C ICS         CONTROL SYSTEM TYPE:  RATE COM/ATT. HOLD
C                                 PURE RATE COMMAND
C                                 ATT. COM./ATT. HOLD.
C MCS       DIRECTIONAL CONTROL SYSTEM.
C ICONFI    CRAFT CONFIGURATION:  UH60, AH1S OR LHX.
C
C
C******************************************************
C
          INCLUDE '[HELHAC.FOR]HAC1.INC/LIST'
                INCLUDE '[HELHAC.FOR]HAC2.INC/LIST'

C
C******************************************************
```

```
C*****************   EXECUTABLE  CODE   ***************
C*****************************************************
C
C------------    BEEPER TRIM DRIVE   ---------------
C
C------DISCRETE TRIM BEEPERS ON CYCLIC INTEGRATE
C------TRIM POSITIONS
C
       TRIM1C = TRIM1C + GRLD*(IRWD - ILWD)*DT2
       TRIM2C = TRIM2C + GNUD*(IANU - IAND)*DT2
C
C
C
C
C--------------------------------------------------
C--------            MAG BRAKE          ------------
C
C------ IMBC IS MOMENTARY SWITCH ON CYCLIC
C------ WHICH CAUSES CONTROL FORCES TO LATCH
C------ ON OR OFF
C
C
C-----IF MAG BRAKE BUTTON DEPRESSED FIRST TIME THIS
C-----CYCLE, TOGGLE MODE
C
      IF( (IMBC .EQ. 1) .AND. (IMBCP .EQ. 0) )
     .       IMB = 1 - IMB
      IMBCP = IMBC
C
C
C-----FLOATING MAG BRAKE VARIABLE
C
      FIMB = IMB
C
C
C-----IF MAG BRAKE ENGAGED, SET TRIM POINT
C-----TO CURRENT POSITION(NO APPARENT FORCE)
C
      IF(IMB .EQ. 0) GO TO 930
            TRIM1C = ROLLOP
            TRIM2C = PITCHOP
            TRIM3C = YAWOP
 930  CONTINUE
C
C
C
C-----------------------------------------------------
C
C
C
     CONTINUE
       RETURN
       END
```